User Manual
GEFSOC Soil Carbon Modeling System

Mark Easter[1], Keith Paustian, Kendrick Killian, Ty Boyack, Steve Williams, Ting Feng, Kevin Coleman[2], Amy Swan, Rida Al-Adamat[3], Tapas Bhattacharrya[4], Carlos E.P. Cerri[5], Peter Kamoni[6], Niels Batjes[7], and Eleanor Milne[8]

Natural Resource Ecology Laboratory
Colorado State University
Fort Collins, CO 80521
http://www.nrel.colostate.edu/projects/agecosys/

[1] Corresponding Author's address: Natural Resource Ecology Laboratory, NREL- NESB Campus Delivery 1499, Colorado State University, Fort Collins, CO 80523-1499. http://www.nrel.colostate.edu/, marke@nrel.colostate.edu
[2] Rothamsted Research Centre, Harpenden, Hertfordshire AL5 2JQ, UK
[3] Higher Council for Research and Development/Badia Research and Development Centre, Safawi, Mafraq Jordan
[4] National Bureau of Soil Survey and Land Use Planning, (ICAR), Amravati Road, Nagpur 440 010, India
[5] Centro de Energia Nuclear na Agricultura, Universidade de Sao Paulo, Av. Centenário, 303 - CEP 13400-961, Piracicaba, São Paulo - Brasil
[6] Kenya Soil Survey, Kenya Agricultural Research Institute, P.O.Box 14733, NAIROBI, 00200, Kenya
[7] ISRIC- World Soils Information, PO Box 353, 6700 AJ Wageningen, the Netherlands
[8] The Department of Soil Science , The University of Reading, Whiteknights, PO Box 233, Reading, RG6 6DW, UK

Table of Contents

Table of Contents 2
Administrative Topics 4
Acknowledgements 4
Warranty and Distribution Requirements 4
Modifications to this Software 5
License 5
Trademarks and copyrights 6
The GEFSOC Soil Carbon Modeling System 7
Introduction 7
Hardware Installation 8
Operating System Installation 9
Redhat® LINUX Operating System 9
Installing the GEFSOC Soil Carbon Modeling System 9
Microsoft® Windows Operating System 10
Installing additional Windows Software 10
Setting up Windows XP and 2000 computers 11
Building Input Datasets for Regional Model Runs with the Graphical User Interface (GUI) 13
Introduction 13
Data Classes 14
Example Dataset 14
Potential Natural Vegetation 15
Land Management 15
Soils 17
Climate 20
Latitude 24
Longitude 25
Other Input Datasets Constructed by the Modeling System 26
Model Run Table 26
Management Sequences 28
Regression Intervals 28
IPCC Method Simulation Runs 29
Preparing Input Data for Simulation Runs 31
Example Dataset 31
Managing Regional Model Runs with the NREL Regional Century Scripting System 36
Introduction 36
Overview 36
System Architecture 37
Management 38
The Run Server 38
Starting the Server 38
Server status 39
Communicating with the Server 40
Stopping the server 41
Debugging simulation and server problems 42

Starting the calculation clients 43
Stopping the calculation clients 44
Debugging the calculation clients 44
Century Errors or gefeq.pl/gefrun.pl trapped errors 45
PERL errors 45
IEEE errors 45
Reading a calculation log file 46
Debugging Crop History problems 48
Running the Century Block Check routine 48
Troubleshooting Errors and Warnings 51
Post-Processing 54
Appendix 1: Redhat® LINUX Fedora Installation Tutorial 54
Appendix 2: Tutorial on using the Graphical User Interface 55
Build Crops 55
Build Rotations 57
Build Histories 59
Build Soils Classification Dataset 61
Climate Data 62
Model Run Table 63
Management Sequences 65
Regression Intervals 67
Generate Files 68
Defaults 68
Generate Files and Run Models Buttons 69
Appendix 3: Modeling System Installation Script for LINUX 71
Appendix 4: Soil Texture Classification Function 75
Appendix 5: Soil Drainage Classification Function 76
Appendix 6: Function defining IPCC soil classifications based on SOTER classifications (I guess this has to be amended to incorporate Niels' suggestions?) 76
Appendix 7: Additional Information Sources 77
Bibliography 78

Administrative Topics

Acknowledgements

The authors gratefully acknowledge the assistance of the following collaborators who provided invaluable suggestions for the GEFSOC modeling system and this document: Martial Bernoux, Carlos Cerri, P. Chandron, Pete Falloon, Christian Feller, Gunther Fischer, Patrick Gicheru, David Jenkinson, Peter Kamoni, C. Mondal, Dilip K. Pal, David Powlson, Zahir Rawajfih, S.K. Ray, Mohammad Shahbaz, Francesco Tubiello, and Stanley Wokabi.

We are especially indebted to Dr Mohammed Sessay of the Global Environmental Facility for his encouragement, guidance, and strong support for this work.

Funding to develop the GEFSOC modeling system and this document was provided through grants from The Global Environmental Facility (administered by The UNEP, Project no. GFL/2740-02-4381) and the co-funders listed below:

Biotechnology and Biological Sciences research Council
Jordan Badia Research and Development Centre
Conselho Nacional De Desenuolimento Cientifico E Tecnologico
Department for International Development
Fundção de Aparo à Pesquisa do Estado de São Paulo
Indian Council for Agricultural Research
International Institute for Applied Systems Analysis
Institut de recherché pour le development
The Hadley Centre
Natural and Environmental Research Council
Natural Resource Ecology Laboratory
Rothamsted Research
System for Analysis, Research and Training
United States of America International Development ??
Universidade de São Paulo
Netherlands Ministry of Housing, Spatial Planning and the Environment

Warranty and Distribution Requirements

Copyright © 2005 by the Natural Resource Ecology Laboratory at Colorado State University. All rights reserved by Colorado State University.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR

PURPOSE. See the GNU General Public License for more details (http://www.gnu.org/copyleft/gpl.html - SEC1).

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA.

If the software is modified by other users and passed on, all recipients must know that what they have is not the original, so that any errors or changes introduced by others will not reflect on the original authors' reputations.

Modifications to this Software

Any and all changes to this software that are included in versions sent to other parties must be provided to the corresponding author listed on the title page to this document. The authors welcome any and all suggestions for improvements, changes, and modifications. Other users are free to modify, improve, or otherwise adapt this software for non-profit educational and scientific purposes that meet the license requirements.

User Group

Colorado State University is sponsoring a user group for this software. Details can be found at http://www.nrel.colostate.edu/projects/agecosys/. The GEFSOC modeling system has been released under an open source license. The aim of open source software is that users may delve into the system and software and offer suggestions for improvements, write improvements of their own, share questions and tips, without depending upon a single user or small group of users to implement improvements. There are two email lists, also called "listservs", to which interested parties may subscribe in order to share information:

General Announcements: This email list provides announcements to users of the GEFSOC system, regarding updates, bug fixes, and general announcements pertaining to related topics. Traffic is expected to be on the order of one to four announcements per month. Any person can join the listserv. The address is GEFSOC_ANNOUNCE-L@colostate.edu. To join the list, do one of the following:

- go to the following web page and select "GEFSOC_ANNOUNCE-L" from the list of available mailing lists: http://www.acns.colostate.edu/?page=frame&src=http%3a%2f%2fwww.colostate.edu%2fServices%2fACNS%2flistserv%2fsubother.html
- Or, you can send an email to listproc@colostate.edu with the words "subscribe GEFSOC_ANNOUNCE-L" in the first line in the email, and no words in the subject line (leave the subject line completely blank)

This list is moderated, which means that postings to the list must be approved by a single moderator. Postings of general interest to interested GEFSOC parties will be accepted.

GEFSOC Users: This email list is intended to be a discussion list for users of the GEFSOC system. Users may share tips, improvements, describe software bugs, ask questions, etc. All subscribers are urged to contribute to discussions shared in the list. The amount of email traffic will be dependent upon the degree of interest. Any person can join the listserv. The address is GEFSOC_USERS-L@colostate.edu. To join the list, do one of the following:

- go to the following web page and select "GEFSOC_ANNOUNCE-L" from the list of available mailing lists: http://www.acns.colostate.edu/?page=frame&src=http%3a%2f%2fwww.colostate.edu%2fServices%2fACNS%2flistserv%2fsubother.html

- Or, you can send an email to listproc@colostate.edu with the words "subscribe GEFSOC_USERS-L" in the first line in the email, and no words in the subject line (leave the subject line completely blank).

The list is unmoderated, which means that any user subscribed to the list may post an email to the list at any time.

Further details on using listserv tools is available at http://www.colostate.edu/Services/ACNS/listserv/docs/usercard.html.

License

This software is provided free of charge to non-profit educational and scientific organizations conducting research on soil carbon and greenhouse gases in the public interest.

Any use of this software for profit-making enterprises is precluded without a negotiated software license from Colorado State University. Users interested in using this software in profit-making enterprises must contact the corresponding author listed on the title page of this document before beginning any such enterprise. Any distribution of this software or modified versions thereof for purposes other than non-profit educational or scientific reasons will be considered to be copyright infringement and a violation of the licensing agreement.

Trademarks and copyrights

Microsoft® is a trademark of the Microsoft Corporation.
Redhat® is a trademark of the Red hat, Inc.
ActiveXperts® is a trademark of ActiveXperts Software B.V.
Century© is copyrighted by the Natural Resource Ecology Laboratory, Colorado State University.
RothC© is copyrighted by Rothamsted Research
ESRI® is a trademark of the ESRI Corporation.
Textpad© is copyrighted by Helios Software Solutions, Inc.

The GEFSOC Soil Carbon Modeling System

Introduction

The GEFSOC Soil Carbon Modeling System was assembled under a co-financed project supported by the Global Environmental Facility (GEF)(administered by The UNEP, Project no. GFL/2740-02-4381)[i]. The system was built to provide scientists, natural resource managers, policy analysts, and others, with the tools necessary to conduct regional- and country-scale soil carbon inventories. It is intended to allow users to assess the affects of land use change on soil carbon stocks, soil fertility and the potential for soil carbon sequestration.

This tool was developed in conjunction with case-study analyses conducted by scientists living in four countries: Brazil, Jordan, Kenya, and India. The analyses were conducted on the Brazilian Amazonian region, the entire countries of Jordan and Kenya, and the Indo-Gangetic plain of India. As such, it has been tested as a proof of concept in developing tropical and arid countries with diverse land use patterns and pressing problems with agricultural soils fertility.

The tool requires a minimum of two off-the-shelf computers with basic networking capabilities and utilizes, to the maximum extent possible open-source software that is freely available for downloading off the web.

The tool conducts this analysis using three well-recognized models and methods:

- The Century general ecosystem model[ii] [iii].
- The RothC soil carbon decomposition model[iv].
- The Intergovernmental Panel on Climate Change (IPCC) method for assessing soil Carbon at regional scales[v].

This tool is intended to interact with a SOTER[vi] database that has been constructed for the country or region the user intends to model.

The hardware necessary for this system is shown in Figure 1.

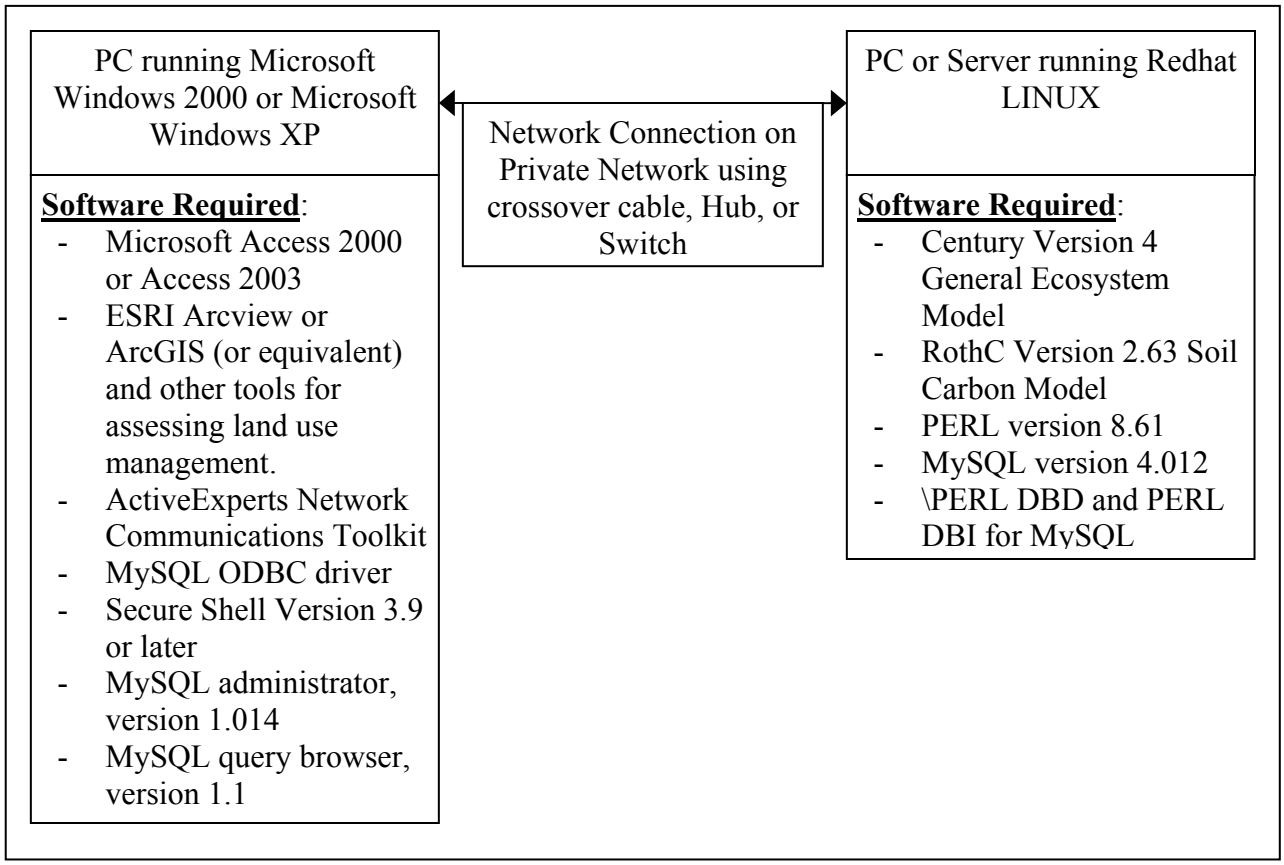

PC running Microsoft Windows 2000 or Microsoft Windows XP
Software Required:
- Microsoft Access 2000 or Access 2003
- ESRI Arcview or ArcGIS (or equivalent) and other tools for assessing land use management.
- ActiveExperts Network Communications Toolkit
- MySQL ODBC driver
- Secure Shell Version 3.9 or later
- MySQL administrator, version 1.014
- MySQL query browser, version 1.1
Network Connection on Private Network using crossover cable, Hub, or Switch
PC or Server running Redhat LINUX
Software Required:
- Century Version 4 General Ecosystem Model
- RothC Version 2.63 Soil Carbon Model
- PERL version 8.61
- MySQL version 4.012
- \PERL DBD and PERL DBI for MySQL

Figure 1. Hardware required for the GEFSOC modeling system.

Hardware Installation

The PC system is recommended to have adequate speed, memory, and hard drive capacity to run Microsoft® Access 2000/2003 and the ESRI® GIS software. The LINUX computer is recommended to be a workstation- or server-class machine with a minimum of 1 GB of RAM and 80 GB of hard drive space. Optional but useful features include multiple high-speed hard drives configured in a RAID 3 or RAID 5 configuration and two or more on-board processors with high speed cache memory. The interconnection between the two computers is recommended to be a category 5 crossover cable or a quality switch. Low-cost hubs are generally not effective network interconnection equipment for equipment of this type. If the PC is also connected to a local area network, then it must be configured with two network cards, one for the local area network, and one for the private GEFSOC network.

Operating System Installation

Redhat® LINUX Operating System

At the time of this writing, we recommend Redhat® Linux Fedora utilizing the Core 2 distribution, available for download either from Redhat® or from the NREL GEFSOC project web site (www.nrel.colostate.edu/projects/agroecosystems/gefsoc.html). A tutorial with recommended installation settings for the system is provided in *Appendix 1: Redhat® LINUX Fedora Installation Tutorial*.

Installing the GEFSOC Soil Carbon Modeling System

After the LINUX operating system is installed, the GEFSOC Soil Carbon Modeling System software packet for LINUX must be downloaded from the NREL GEFSOC project website and installed on the LINUX computer. This process installs:

- PERL release 8.61
- MySQL version 4.012
- PERL DBD
- PERL DBI for MySQL
- rsh daemon version 0.17 release
- telnet daemon version 0.17 release
- GEFSOC soil carbon modeling system.

It also sets SAMBA and NFS shares to be compatible with the needs of the PC GUI. A summary of those settings follows:

- SAMBA:
  - Browseable share to /usr/local/nrel/ defined as “usrlocalnrel”.
  - Browseable share to /home/gefsoc/ defined as “gefsoc”
  - Browseable share to /usr/local/mysql/var defined as “usrlocalmysqlvar”

- NFS:
  - Share to /usr/local/nrel
  - Share to /home/gefsoc
  - Share to /usr/local/mysql/var

- Built-in firewalls in the LINUX system must be shutdown completely

The installation process is as follows:

1. Login to the LINUX computer as root
2. Open a terminal window on the LINUX computer either from the LINUX X11 GUI or using secure shell from the PC.
3. Copy the file GEFSOC_LINUX.tar.gz from the installation CD to /root
4. Execute the following commands:

   cd /root
   gunzip ./GEFSOCM.tar.gz

tar –Pxvf ./GEFSOCM.tar
./install_GEFSOCM
smbpasswd - gefsoc

The final command (smbpasswd –a gefsoc) will ask the user to type in the new password (twice) associated with the SAMBA gefsoc account. The user should use the password 'gefsoc'.

This installation process could take up to 30 minutes depending on the speed and memory capacity of the LINUX server. The script that is installed is shown in Appendix D.

Users are advised after the installation script is run to manually confirm the above SAMBA and NFS settings using the LINUX X11 graphical user interface, and also confirm that the SAMBA, NFS, and MySQL services are set to run automatically upon bootup as specified in the "services" manager of the LINUX operating system.

Microsoft® Windows Operating System

The Windows operating system must be Windows 2000 running service pack 4 or Windows XP running the most recent service packs and updates, with one exception: Windows XP service pack 2 is not recommended at the time of this writing due to network configuration issues.

If a third-party firewall or the built-in firewall in Windows XP is used, it must either be disabled for the private network or configured specifically to allow unimpeded IP communications between the PC and the LINUX computer. This includes rsh, rlogon, and rexec commands issued from the ActiveXperts® software on the PC to the LINUX computer, and TCP/IP responses from the LINUX server back to the PC.

Installing additional Windows Software

Users will need to install four additional software packages. The software packages include:

- Secure Shell capable of running ssh2 protocols (free under the GNU public license).
- ActiveXperts® Network Communications tool (universal site license already been purchased for this software).
- MySQL ODBC driver, version 3.51 (free to non-profit educational and research organizations under an open source license from MySQL AB).
- MySQL administrator for managing the MySQL server on the LINUX computer (free to non-profit educational and research organizations under an open source license from MySQL AB).
- MySQL query browser for managing MySQL queries on the LINUX computer (free to non-profit educational and research organizations under an open source license from MySQL AB).
- Textpad© text file editor version 4.0 or equivalent, by Helios Software Solutions, PO Box 619, LONGRIDGE, PR3 2GW, England. Tel: +44 (1772) 786373, Fax: +44 (1772) 786375, Web: http://www.textpad.com

Setting up Windows XP and 2000 computers

Users must configure their Windows XP and 2000 computers to meet the following criteria:

- **Microsoft® Access References:** The GEFSOC GUI must be configured under Microsoft® Access to have the following references turned on:
  - Visual Basic for Applications
  - Microsoft® Access 10.0 Object Library
  - Microsoft® Visual Basic for Applications Extensibility 5.3
  - Microsoft® DAO 3.6 Object Library
  - ActiveSocket 2.3 Type Library
  - OLE Automation

  To turn on these references, double-click on the "Global" module in the "Modules" tab of the database window. Then select the "Tools" menu, and then "References", and then make sure that the above references are checked.

- **Firewalls:** Users running Microsoft® Windows XP must have all firewalls turned off for the private network used for the GEFSOC project.
- **Macro Security:** Users running Microsoft® Excel 2000 or XP must have macro security set to medium in order to use macros provided with the included soil classification spreadsheet. When the spreadsheet is opened, they must select the "enable macros" option. Otherwise, with macro security set to high or with macros otherwise disabled, the soil classification function will not work correctly.
- **Mapped Network Drives:** There must be three network drive shares set up on the PC that are mapped to SAMBA shares on the LINUX server. Those shares are:
  - \\gefsoc\gefsoc Suggested mapped drive letter is "y:\". These are the files stored on the LINUX server at /home/gefsoc/. This directory contains the basic data files and scripts necessary to run the modeling system.
  - \\gefsoc\usrlocalnrel Suggested mapped drive letter is "z:\". These are the files stored on the LINUX server at /usr/local/nrel/. This directory contains the perl modules necessary to run the modeling system.
  - \\gefsoc\usrlocalmysqlvar Suggested mapped drive letter is "x:\". These are the mysql database tables stored at /usr/local/mysql/var. The files are those used by the modeling system GUI.
- **Network Settings and IP Addresses:** The PC on the private network must be configured with an IP address compatible with the LINUX computer. The REQUIRED settings for the TCP/IP protocol are an IP address of 10.10.11.101, a net mask of 255.255.255.0, with DNS server settings left blank.
- **ODBC Data Source Name:** The PC must have an ODBC ***System*** Data Source Name (DSN) that can be used to link the GUI to the LINUX MySQL database. The DSN must have the following settings:
  - Data Source Name: mysql gefsoc
  - Server: 10.10.11.101
  - User: gefsoc
  - Password: gefsoc
  - Database: gefsoc

- Allow Big Results: set to true (checked)
- Change Bigint columns to Int: set to true (checked)

Building Input Datasets for Regional Model Runs with the Graphical User Interface (GUI)

Introduction

There are six basic data classes required to build the datasets necessary for a regional simulation. They include:

- **Potential Natural Vegetation**. In order to initialize The models (hereinafter referred to as "the models"), it is necessary to run the models according to equilibrium conditions that existed prior to the beginning of historic human land use.
- **Historic and Current Land Use Management**. Modeling the historical land use for 100 years before present is necessary for an accurate assessment of current soil C stocks and change rates[vii] [viii]. This modeling system relies upon the user assembling historic and current block sequences of management activities and defining the area proportion relationship between these sequences according to historic and/or predicted transition rates between land management systems.
- **Climate**. Climate information, either grid-based, polygon based, or point based (from weather stations), is necessary for The models and the IPCC method.
- **Soils**. The models require basic physical information on soil texture and bulk density. The IPCC method requires that soils be classified according to texture and/or general physico-chemical activity. A SOTER database for the modeling region and associated GIS coverages is required. If creating a new SOTER database for the region to be modeled, it is not necessary to complete all of the terrain landuse and climate info in the SOTER database. Only the soils info is important.
- **Latitude**. The models require the Latitude of the region being modeled.
- **Longitude**. The models require the Longitude of the region being modeled.

We recommend that these data be assembled into a GIS with separate coverages for each parameter. An example of these coverages from a regional model run in the Great Plains of North America is described starting on page 14 in the section titled *Data Classes*. The data classes are organized into a single relational database with several tables. The relationships between these tables are shown in Figure 2.

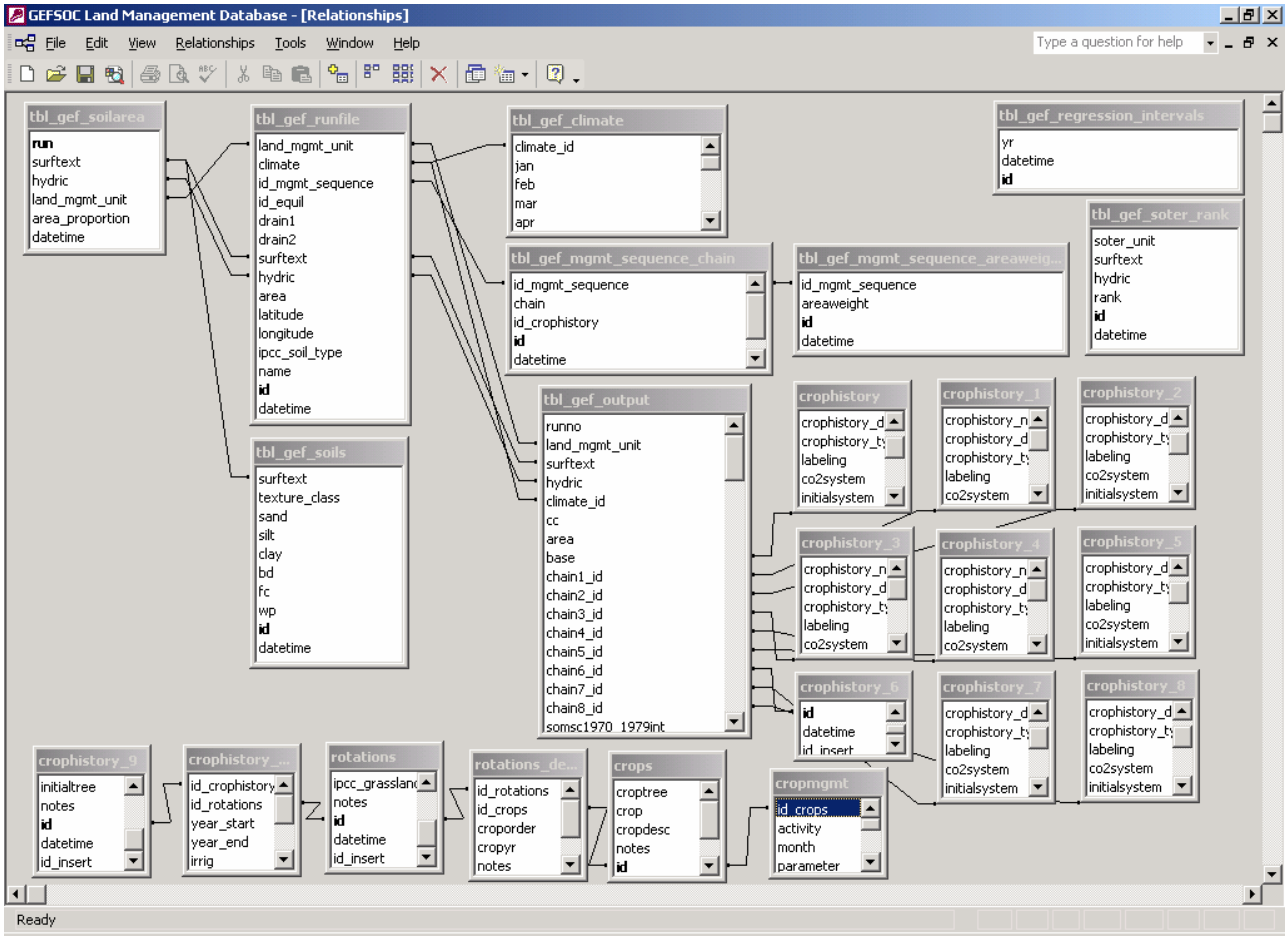


Figure 2. GEFSOC database relationships

Data Classes

Example Dataset

For instructional purposes, we have provided an example dataset from the Northern Great Plains of the U.S. This dataset consists of two major land resource areas (MLRAs) from Montana, the Westernmost (MLRA 58A) of which is in the Rocky Mountain foothills and consists largely of ponderosa pine grassland savannahs, on steep slopes unsuitable for tillage agriculture, and the Easternmost consisting of mixed grass prairie where there is a mixture of tillage agriculture for annual grains and perennial forage crops, as well as grazing on native grasslands.

There are six datasets for which GIS coverages must be built:

- Potential natural vegetation
- Base land management polygons
- Climate polygons
- SOTER soils polygons
- Latitude of base land management polygons

- Longitude of base land management polygons

Examples of each of these are described in the following sections. Users are referred to *Appendix 2: Tutorial on using the Graphical User Interface* for details on using the graphical user interface.

Potential Natural Vegetation

A GIS coverage of potential natural vegetation (PNV) is necessary for the GEFSOC modeling system. Equilibrium crops, rotations, and histories must be constructed in the GUI to correspond to the types of vegetation types defined in the coverage. The name of each potential natural vegetation type must correspond exactly to the equilibrium crop history name defined in the GUI for each region. The PNV data are written to the Run Table at the time that these data coverages are overlain to produce an intersection of the four datasets.

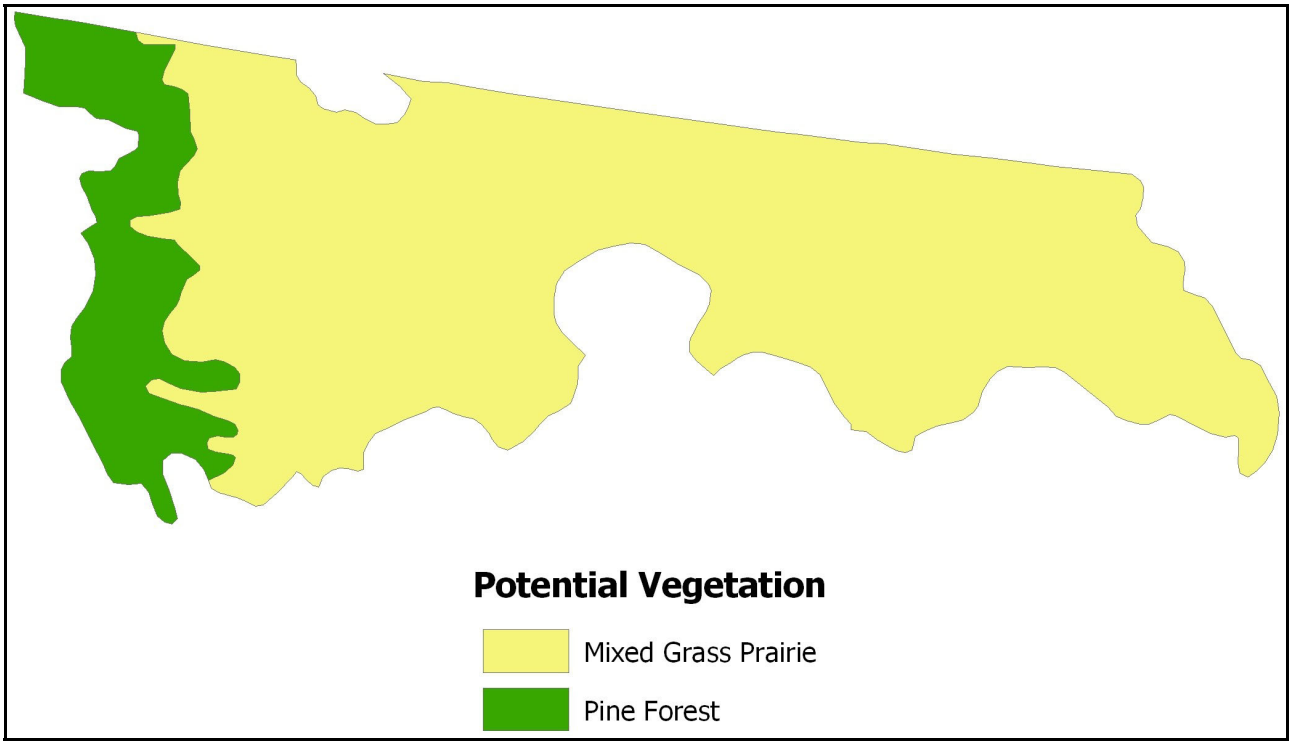


Figure 3. Potential natural vegetation for the example modeling region. Mixed grass prairie is modeled by modeling a G4 grass (75% cool season and 25% warm season grasses) with periodic low-intensity grazing and a seven year fire return interval. The Pine Forest is modeled by modeling a CONIF tree with a G4 grass as a savannah, with a 30 year fire return interval.

Land Management

The models and the IPCC method are primarily driven by land use, though in different ways. Century is a general ecosystem model that requires specific information at monthly time steps on land management activities like planting, tilling, fertilizing, and harvesting crops, grazing forage, and cutting or thinning forests. The RothC model requires monthly information on carbon inputs and tillage, and the IPCC method requires generalized land management information according to specific land use and land management classes[ix]. Within this modeling system, Carbon inputs and tillage data simulated by the Century model are used to drive the RothC model. Land use

management is classified according to IPCC guidelines, and these classifications are used to drive the IPCC method.

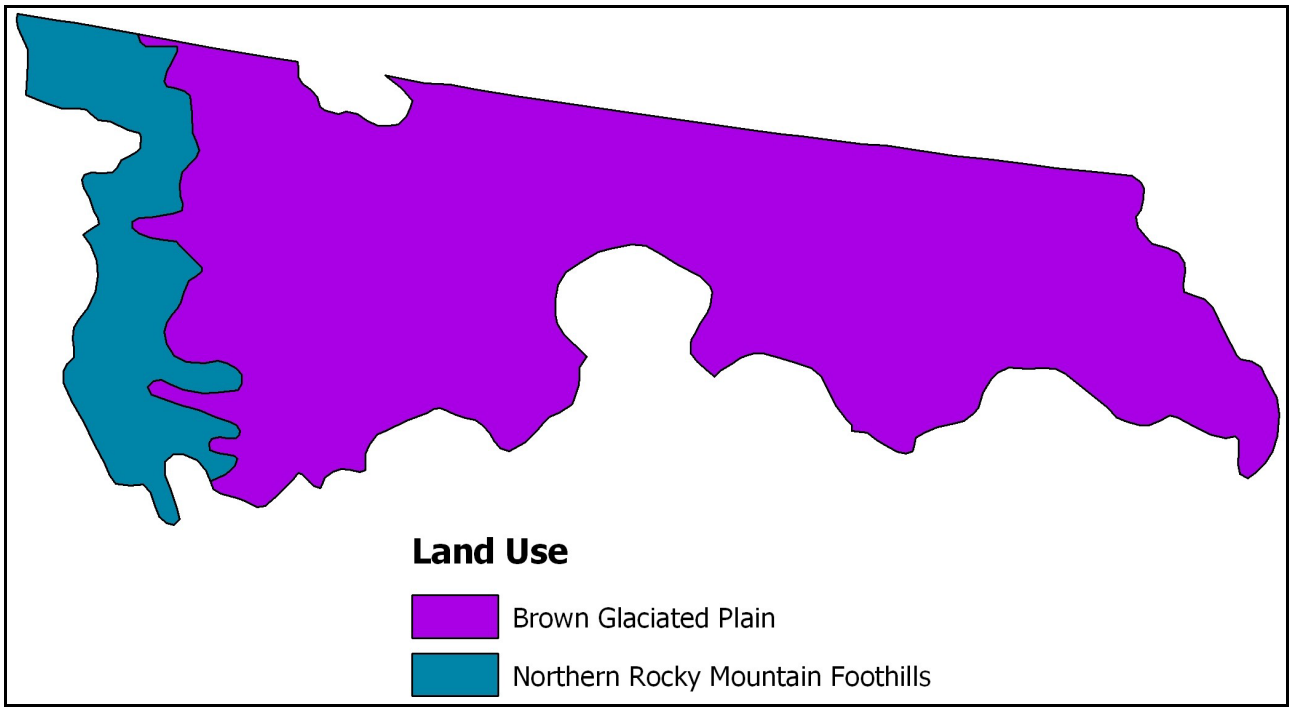


Figure 4. Land management regions for the example dataset. The Northern Rocky Mountain Foothills region is managed for grazing and timber harvesting, and the Brown Glaciated Plain is managed for annual grains, forage crops, and grazing.

There are five basic elements to the Land Management class of data for the modeling system:

- **Events**. These correspond to the monthly events that drive the Century model, including management activities like fertilization, tillage, planting, harvesting, grazing, and cutting and thinning trees. The time step for Events is ***monthly***.

- **Crops and Trees**: The events described above are organized into sequences of management activities associated with certain crops defined in the Century crop.100 file. As such they reference the input parameters from the crop.100 necessary to grow crops and trees in the Century model and the events that describe the land management for the particular crop or tree. The time scale for Crops and trees is generally 1 year for annual crops to several decades for tree fruits.

- **Rotations**. Crops and trees are organized into sequences to make rotations. For example, a fallow-spring wheat system from the example dataset is a rotation of two "crops" (as defined by Century), a fallow crop one year followed by a spring wheat crop the second year. Users familiar with Century will recognize that rotations correspond to "blocks" in the event files. The time scale for rotations is generally one year for annual crops to a decade or more for complex cropping rotations or for tree fruits.

- **Histories**. Rotations are organized into sequences to make blocks of histories. For example, the base history in the example dataset is organized into three sequences of rotations spanning the base period. Each different rotation uses different crops of increasing productivity as time moves from the beginning to the end of the base period, with corresponding changes in fertilizer and tillage practices. The time scale for histories is generally several years to several decades.

- **Management Sequences**. Histories are organized into management sequences. In order to simulate soil C stocks and change rates on a regional scale, it is important to understand that land in any current land management condition may have followed very different past sequences of land management to reach that current state. For example, one field that is currently in irrigated continuous corn may have been converted over from a bottomland hay field after drainage tiles were installed just 20 years ago, and is gravity-irrigated with surface water from a ditch. This management sequence would likely have a base history consisting of pasture and hay management, followed by pasture and hay in the recent period and then a conversion to irrigated corn for the current period. This management sequence would involve a string of three different histories. An adjoining field that is also in irrigated continuous corn may have been native prairie managed as pasture up until twenty years ago, when the landowner drilled a wall and installed a pump-driven center-pivot irrigation system. This field would have a base history of grazing, followed by a recent period of grazing and a current period of irrigated continuous corn. In summary, whereas both are in irrigated continuous corn now, both are likely to have very different soil C stocks and change rates. The time scale for management sequences is generally several decades to a century or more.

Appendix 2: Tutorial on using the Graphical User Interface contains detailed information on how to construct land management information that is required to drive the model. Users should become familiar with the models and the IPCC method in order to achieve the best success[x].

Soils

The SOTER database provides information on soil texture, drainage class, and bulk density for up to ten soil subclasses in each SOTER unit. Within the SOTER database, the subclasses are listed with their relative proportional areas within the SOTER unit in the table 'SOTERunitComposition'. Surface texture, drainage classification, and bulk density for various soil depths are listed in the table 'SOTERparameterEstimates'. The relationship between these tables is shown in Figure 6.

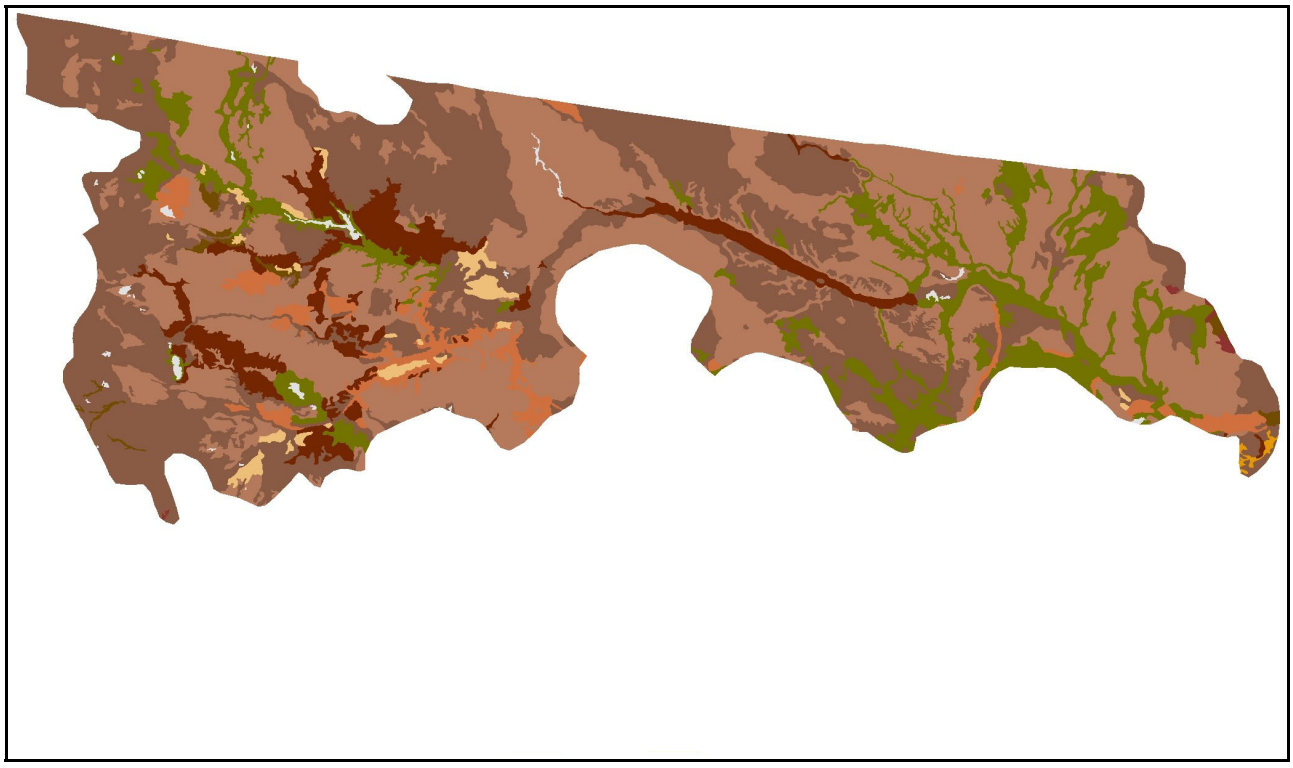


Figure 5. Soils from the SOTER database for the example dataset. There are 148 SOTER units described in this system.

Two database tables are used from the SOTER database in the modeling system: 'SOTERunitComposition', and 'SOTERparameterEstimates'. The relationship between these tables, as used in the modeling system, are shown in Figure 6.

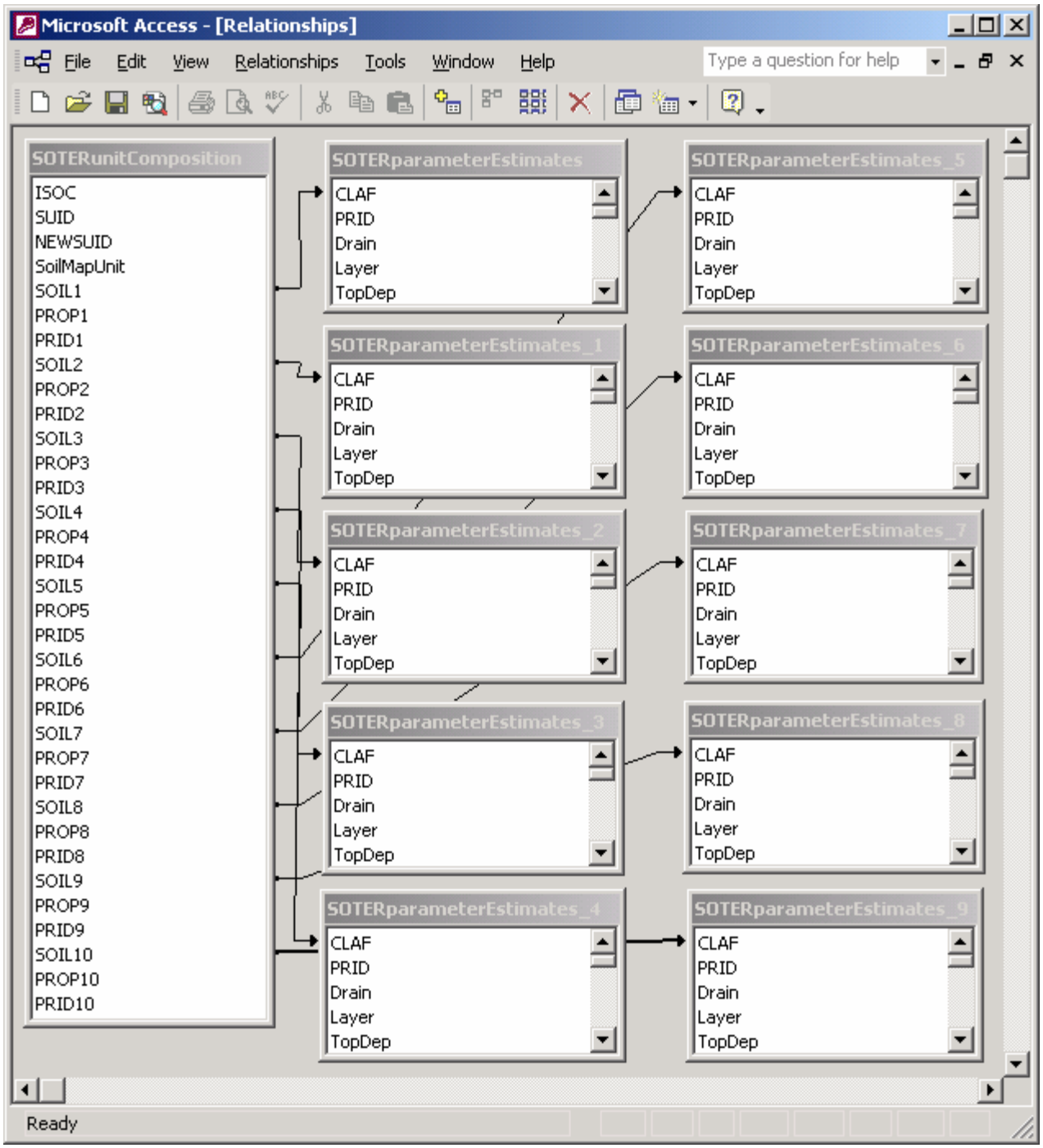


Figure 6. Relationship between SOTER database tables for physical parameters required by the GEFSOC modeling system. Note that there is only one table named 'SOTERparameter' estimates (though from the diagram there may appear to be ten), but it is linked to 'SOTERunitComposition' ten times, once for each of the possible soil types in 'SOTERunitComposition'.

These data are the primary soil physical parameters that are required by the models. Modeling every one of these soils, however, is very time consuming and generally unnecessary. By classifying soils of similar texture within a classification scheme, users can greatly reduce the

number of model runs required while maintaining a relatively high degree of precision and accuracy in the model output. A lookup table is automatically built by the modeling system using the classification scheme described in *Appendix 4: Soil Texture Classification Function.* Users can specify the number of classes within each soil fraction that they wish to use on the "Generate Files" page of the GUI.

Many have asked why they shouldn't just use the classifications within the soil texture triangle in order to classify their soils for this system. Whereas the soil texture triangle is an excellent descriptive tool, it is a qualitative classification scheme that can introduce nonlinear effects when used with the models. This is because the classification scheme within the texture triangle is not linear- for example, the range for the clay fraction within soils classified as clay loam is significantly different from the range for the clay fraction within sandy loam soils. The clay and silt fractions are used as linear multiplication parameters within the models, and hence a linear classification scheme is called for. The function shown in *Appendix 4: Soil Texture Classification Function*, classifies soils linearly according to equal sized ranges defined by the user.

The soils data are organized into a lookup table containing all of the unique soil surface texture classifications used in the model runs. The table definition is shown in Table 1.

Table 1. Field definitions for the SOTER soils data table.

Table tbl_gef_soils		
Field Name	**Field Definition**	**Description**
surftext	char(21) indexed	this classification code is used to link to the model run table in order to provide texture information to the models
sand	float	sand fraction
silt	float	silt fraction
clay	float	clay fraction
id	integer auto_increment primary key	Keyfield
bd	float	bulk density
datetime	timestamp	timestamp field for Microsoft® Access interoperability

Climate

The models and the IPCC method all require basic information on climate in the region to be modeled. Mean monthly precipitation (cm), maximum temperature (ºC), and minimum temperature (ºC) are required for the models, and IPCC climate classes must be determined according to the IPCC good practice guidelines[xi]. Polygons must be built within a GIS that classify climate according to similarity in precipitation and temperature. In general, we recommend that users construct their climate polygons identically to their base land-use polygons, in order to reduce the complexity of the model runs, using point-source data (actual data from weather stations) using spatial statistics, or using grid-based climate such as the U.S. PRISM dataset or other international datasets[xii]. Where there is significant climate variability within the polygons (as described in the following section), climate polygons should be split.

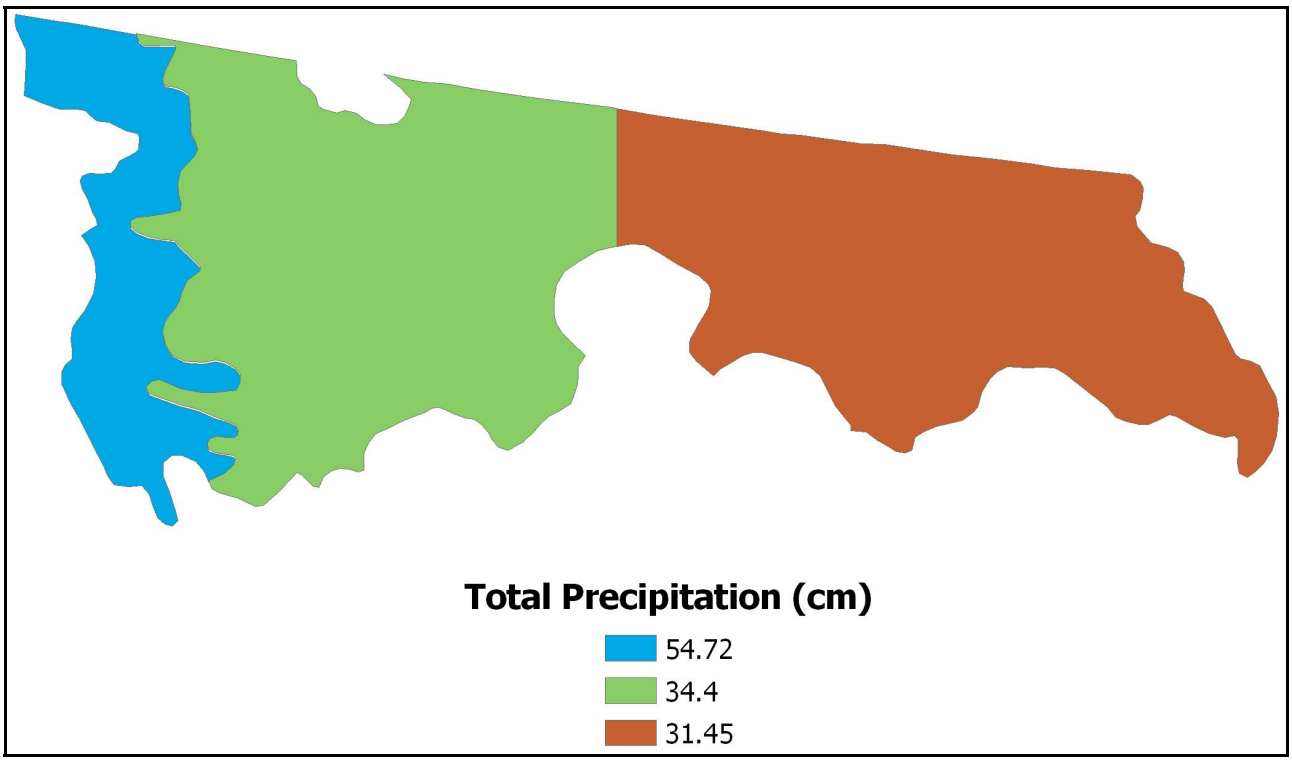


Figure 7. Climate data for the example dataset. Note that the region of the Brown Glaciated Plain has been split into two climate regions, primarily due to variations in temperature within the polygon. The IPCC classification for all three of these regions is "cool temperate, dry".

There are two general approaches that users should consider using when defining climate polygons.

- Where base land use polygons are based on political or municipal boundaries that cross major climate regions, polygons should be constructed using mean total yearly precipitation on 5cm isoclines as the primary criterion, with each climate class identified by a unique alphanumeric identifier. After these polygons are constructed, users must construct data to insert into the climate lookup file described below. An example of this approach is shown for Jordan in Figure 8.
- Where base land use polygons are based on boundaries that do not cross major climate regions, a climate lookup file should be constructed that uses the base land use polygons as its boundaries and uses the base land use polygon identifiers as the climate polygon identifiers. An example of this approach from the Brazilian Amazon is shown in Figure 9.

This consists of a table containing data on mean monthly precipitation, mean monthly maximum temperature, mean monthly minimum temperature and the IPCC climate classification for the unique climate regions used in the model runs It is used as a lookup table by the model runs, to provide climate data to the models and the IPCC method. The table definition is shown in Table 2.

Table 2. Field definitions for the climate data lookup table.

Table tbl_gef_climate		
Field Name	**Field Definition**	**Description**
climate_id	char(60) indexed	this classification code is used to link to the model run table in order to provide texture information to the models
jan	float	January mean total precip, mean tmax, or mean tmin
feb	float	February mean total precip, mean tmax, or mean tmin
mar	float	March mean total precip, mean tmax, or mean tmin
apr	float	April mean total precip, mean tmax, or mean tmin
may	float	May mean total precip, mean tmax, or mean tmin
jun	float	June mean total precip, mean tmax, or mean tmin
jul	float	July mean total precip, mean tmax, or mean tmin
aug	float	August mean total precip, mean tmax, or mean tmin
sep	float	September mean total precip, mean tmax, or mean tmin
oct	float	October mean total precip, mean tmax, or mean tmin
nov	float	November mean total precip, mean tmax, or mean tmin
decr	float	December mean total precip, mean tmax, or mean tmin
total	float	total/mean for year
indicator	enum	“prec” or “tmax” or “tmin”
ipcc_climate_region	char(26)	IPCC climate region classification
id	integer auto_increment primary key	Keyfield
datetime	timestamp(14)	timestamp field for Microsoft® access interoperability

In some cases, defining climate based on base land use polygons is not appropriate. This is generally the case in areas where precipitation and/or temperature are highly variable over short distances. Where this is the case, users may wish to use spatial statistics within their GIS to construct climate polygons based on data from actual weather stations. An example of one such dataset is shown in Figure 8.

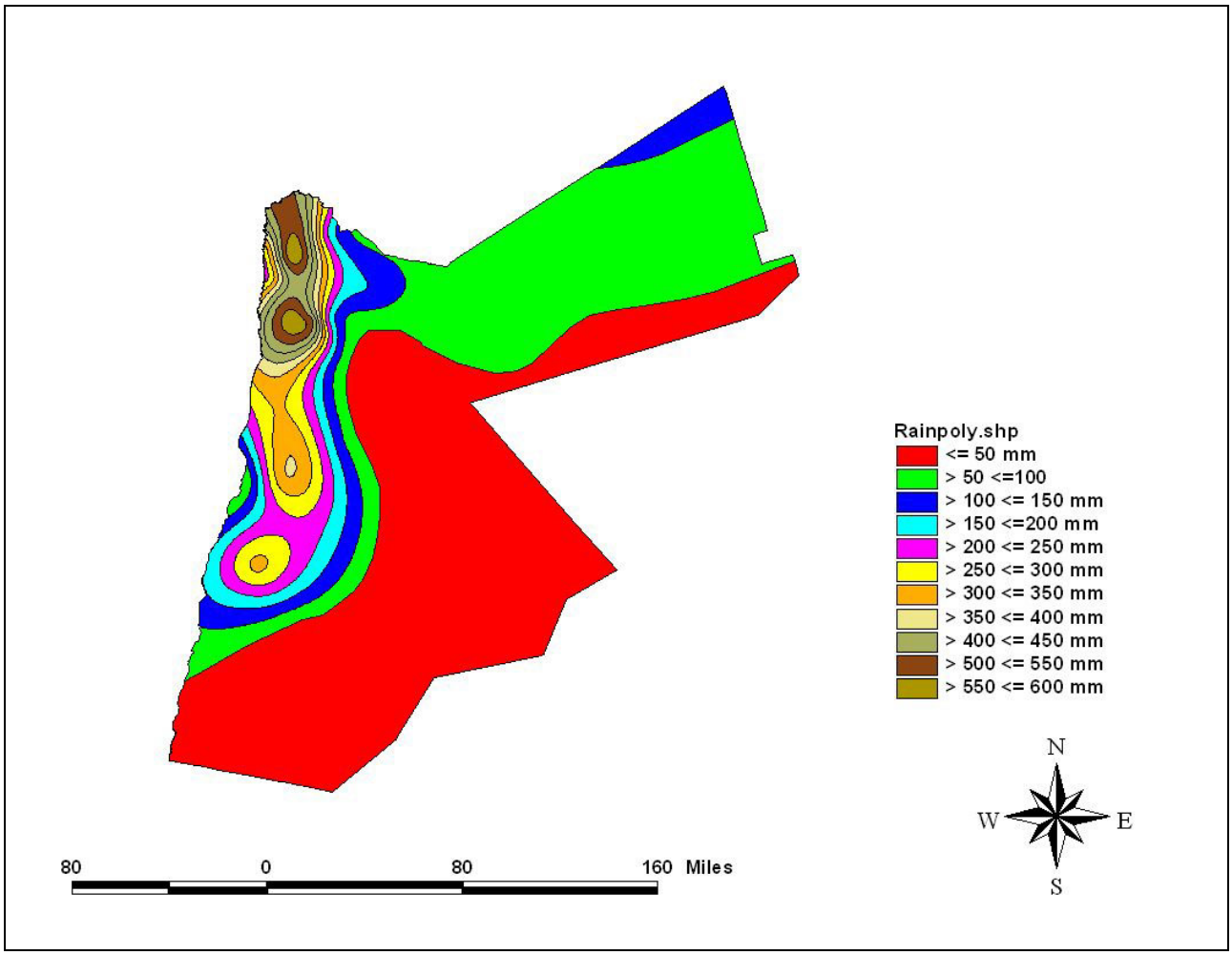


Figure 8. Climate coverages for Jordan.

Figure 9. Climate coverages for the Brazilian Amazon.

Latitude

Latitude is used in the modeling system within the model run file, where it is provided to the models. The latitude of the centroid of each base land use polygon must be calculated within the GIS or using other means, and attached to each base land use polygon so that it may be exported in the intersecting file.

An example polygon for latitude is shown in Figure 10.

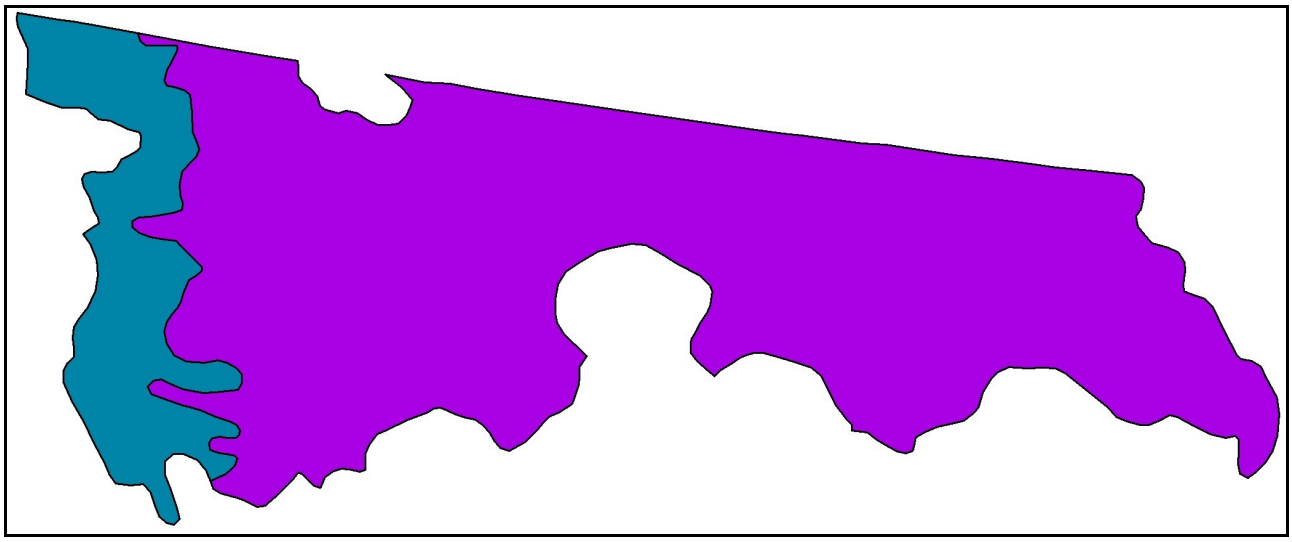


Figure 10. Base land use polygon coverage, containing latitude information for the centroid of each separate polygon.

Longitude

Longitude is used in the modeling system within the model run file, where it is provided to the Century model. The longitude of the centroid of each base land use polygon must be calculated within the GIS or using other means, and attached to each base land use polygon so that it may be exported in the intersecting file.

An example polygon for longitude is shown in Figure 11.

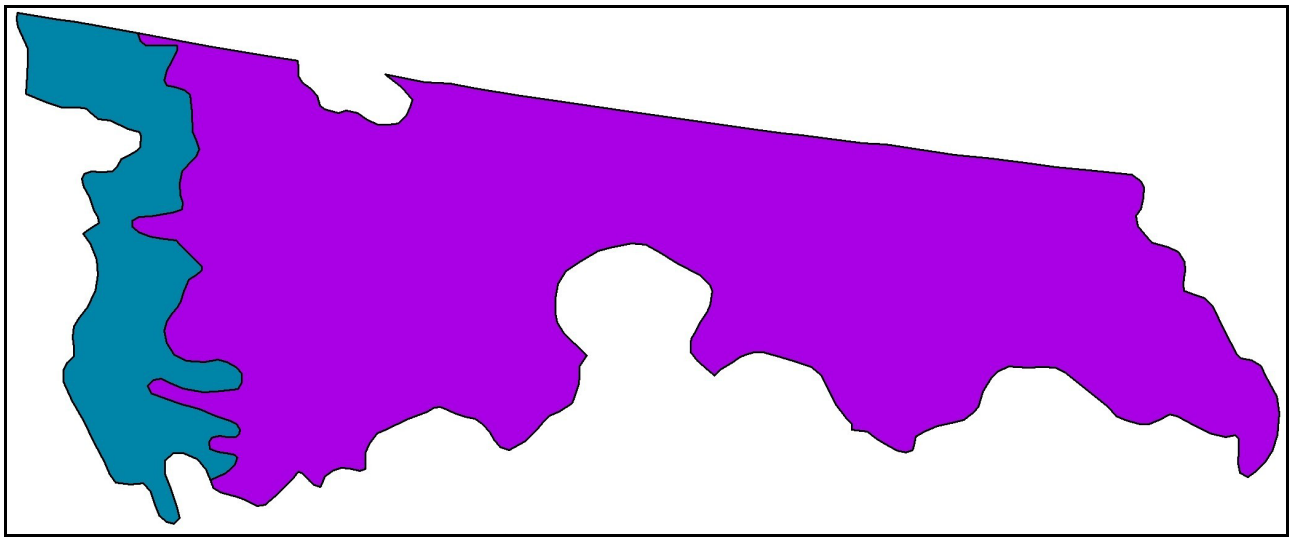


Figure 11. Base land use polygon coverage, containing longitude information for the centroid of each separate polygon.

Other Input Datasets Constructed by the Modeling System

The following datasets must be built for the modeling system to run correctly:

Model Run Table

This consists of a table containing the intersections of the following GIS coverages described starting on page 14 in the section titled *Data Classes*. The table is produced by generating the unique intersection of the six data classes required for the modeling system.

This table joins together all of the information required to run the models for the regional analysis.

Table 3. Runfile table field definitions.

Table tbl_gef_runfile		
Field Name	**Field Definition**	**Description**
land_mgmt_unit	char(50) indexed	Land management unit identifier
climate_id	char(60) indexed	Climate identifier
id_mgmt_sequence	char(60) indexed	Management sequence identifier
id_equil	char(125)	Equilibrium crop history name specified exactly from crop history table
drain1	integer	Median year that ditch drainage was completed
drain2	integer	Median year that tile drainage was completed
soter	char(50)	SOTER NEWSUID
area	float	Area fraction occupied by the intersection of potential natural vegetation, base land use polygon, climate, SOTER unit, latitude, and longitude.
latitude	float	Latitude of the sub-polygon centroid
longitude	float	Longitude of the sub-polygon centroid
Id	integer auto_increment primary key	Keyfield
datetime	timestamp(14)	Timestamp field for Microsoft® access interoperability

The data for the model run table are constructed by overlaying the six GIS coverages, and producing an intersecting dataset. The dataset that results from the intersection may then be output to an Excel worksheet, and manipulated as necessary whereby unneeded columns are deleted and columns are placed in the correct order for pasting into the model run table. This intersection is illustrated in Figure 12.

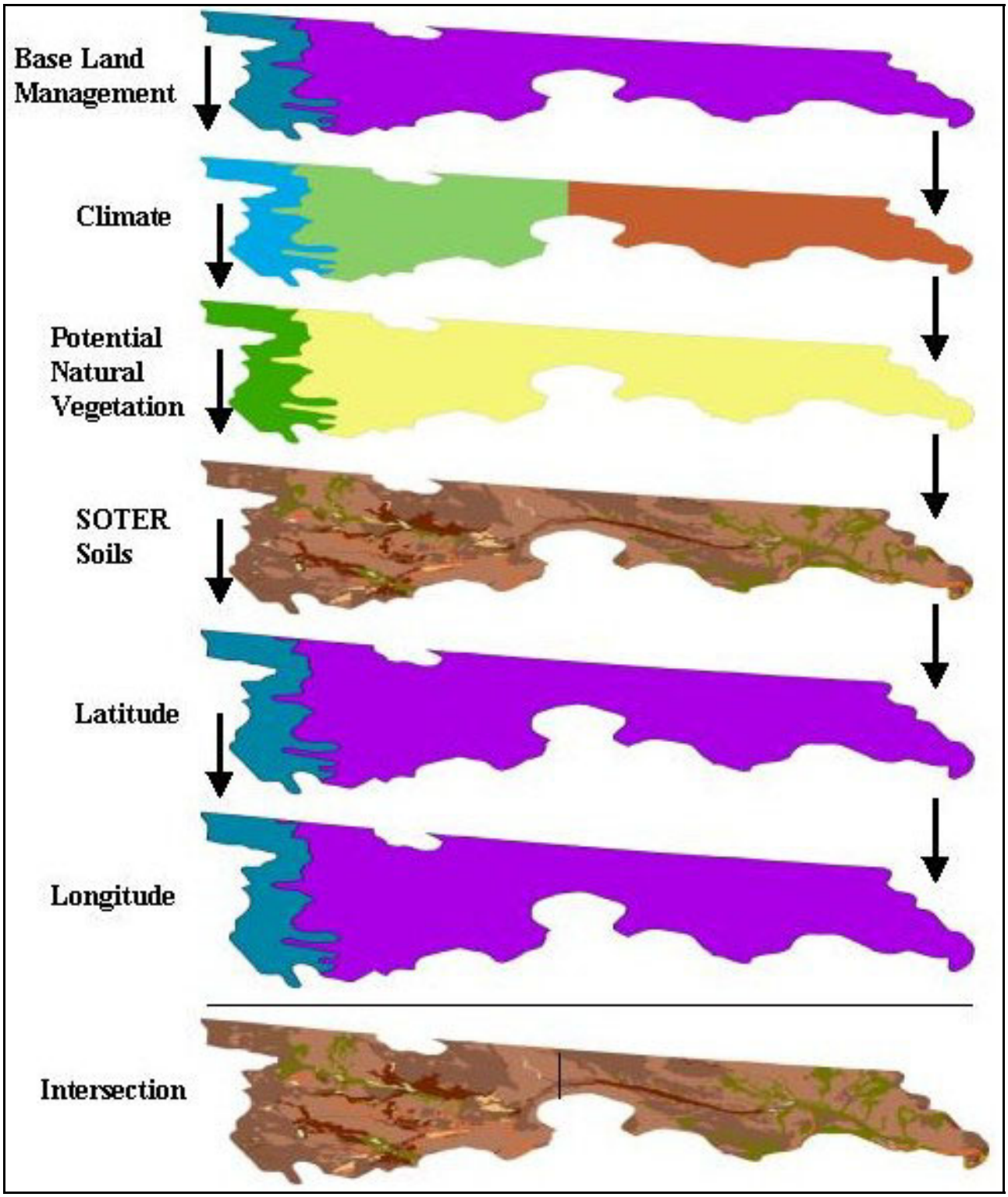


Figure 12. Intersection of the six GIS coverages, used to produce the data for the Model Run Table.

Management Sequences

The management sequences are constructed in the form below, that which stores information in a combination of two tables. The two tables are as follows:

Table 4. Field definitions for the management sequence chain table.

Table tbl_mgmt_sequence_chain		
Field Name	**Field Definition**	**Description**
id_mgmt_sequence	char(50) indexed	management sequence identifier
chain	char(6)	chain in the management sequence
id_crophistory	integer indexed	crophistory id to link to crophistory table
id_crophistory_insert	Integer	retained for future use
id	integer auto_increment primary key	Keyfield
datetime	timestamp(14)	timestamp field for Microsoft® access interoperability

Table 5. Field definitions for the management sequence areaweight table.

Table tbl_mgmt_sequence_areaweight		
Field Name	**Field Definition**	**Description**
id_mgmt_sequence	char(50) indexed	management sequence identifier
areaweight	float	area weighting fraction for this mgmt sequence
id	integer auto_increment primary key	Keyfield
datetime	timestamp(14)	timestamp field for Microsoft® access interoperability

The management sequences are built to link the sequence of crop history blocks associated with the land use histories of a single management sequence. The area proportion associated with each management sequence is entered in the field provided, to be used in post-processing the model run data.

Regression Intervals

The models generate large amounts of output data. A single regional model of run of 50,000 to 100,000 runs requires several gigabytes of storage just to keep the raw data generated by the models. To reduce the size of the dataset while still generating meaningful results for users, the modeling system generates a series of regression statistics for user-defined breakpoints in the management sequence. The user specifies the years for which soil C data are needed in the output dataset, and the modeling system calculates the intercept value, slope and r2 statistic for each time interval between the year breakpoints for the following output data:

- Century somsc
- Century somtc

- Century totc
- RothC soil C

These data are followed by the mean values for each of the following output variables for the time period between breakpoints:

- Century agcacc
- Century bgcacc
- Century cinputs
- Century grain yields

The table format is as follows:

Table 6. Regression interval table field definitions.

Table tbl_regression_interval		
Field Name	**Field Definition**	**Description**
regression_interval	integer	year of breakpoint
id	integer auto_increment primary key	keyfield
datetime	timestamp(14)	timestamp field for Microsoft® access interoperability

The management sequences are built to link the sequence of crop history blocks associated with the land use histories of a single management sequence. The area proportion associated with each management sequence is entered in the field provided, to be used in post-processing the model run data.

IPCC Method Simulation Runs

The soil C stocks and change rates calculated from the IPCC method are calculated using information entered into the land management, climate, and soils data in the GUI. These data are consolidated into a single input table shown in

Table tbl_ipcc_runtable		
Field Name	**Field Definition**	**Description**
land_mgmt_unit	char(50)	Base land use polygon
id_mgmt_sequence	char(50)	Management sequence from the land use polygon
ipcc_soil_type	char(26)	IPCC soil type
ipcc_climate_region	char(21)	IPCC climate region
Area	float	Area associated with this sub-polygon
base_startyr	integer	Year the base history starts
base_endyr	integer	Year the base history ends
base_ipcc_system	char(39)	IPCC land use system for the base history
base_ipcc_grasslandother	char(9)	IPCC grassland/other class for the base history
base_ipcc_mgmt	char(20)	IPCC management class for the base history

base_ipcc_input	char(19)	IPCC Carbon inputs class for the base history
chain1_startyr	integer	Year the chain1 history starts
chain1_endyr	integer	Year the chain1 history ends
chain1_ipcc_system	char(39)	IPCC land use system for the chain1 history
chain1_ipcc_grasslandother	char(9)	IPCC grassland/other class for the chain1 history
chain1_ipcc_mgmt	char(20)	IPCC management class for the chain1 history
chain1_ipcc_input	char(19)	IPCC Carbon inputs class for the chain1 history
chain2_startyr	integer	Year the chain2 history starts
chain2_endyr	integer	Year the chain2 history ends
chain2_ipcc_system	char(39)	IPCC land use system for the chain2 history
chain2_ipcc_grasslandother	char(9)	IPCC grassland/other class for the chain2 history
chain2_ipcc_mgmt	char(20)	IPCC management class for the chain2 history
chain2_ipcc_input	char(19)	IPCC Carbon inputs class for the chain2 history
chain3_startyr	integer	Year the chain3 history starts
chain3_endyr	integer	Year the chain3 history ends
chain3_ipcc_system	char(39)	IPCC land use system for the chain3 history
chain3_ipcc_grasslandother	char(9)	IPCC grassland/other class for the chain3 history
chain3_ipcc_mgmt	char(20)	IPCC management class for the chain3 history
chain3_ipcc_input	char(19)	IPCC Carbon inputs class for the chain3 history
chain4_startyr	integer	Year the chain4 history starts
chain4_endyr	integer	Year the chain4 history ends
chain4_ipcc_system	char(39)	IPCC land use system for the chain4 history
chain4_ipcc_grasslandother	char(9)	IPCC grassland/other class for the chain4 history
chain4_ipcc_mgmt	char(20)	IPCC management class for the chain4 history
chain4_ipcc_input	char(19)	IPCC Carbon inputs class for the chain4 history
chain6_startyr	integer	Year the chain6 history starts
chain6_endyr	integer	Year the chain6 history ends
chain6_ipcc_system	char(39)	IPCC land use system for the chain6 history
chain6_ipcc_grasslandother	char(9)	IPCC grassland/other class for the chain6 history
chain6_ipcc_mgmt	char(20)	IPCC management class for the chain6 history
chain6_ipcc_input	char(19)	IPCC Carbon inputs class for the chain6 history
chain7_startyr	integer	Year the chain7 history starts
chain7_endyr	integer	Year the chain7 history ends
chain7_ipcc_system	char(39)	IPCC land use system for the chain7 history
chain7_ipcc_grasslandother	char(9)	IPCC grassland/other class for the chain7 history
chain7_ipcc_mgmt	char(20)	IPCC management class for the chain7 history
chain7_ipcc_input	char(19)	IPCC Carbon inputs class for the chain7 history
chain8_startyr	integer	Year the chain8 history starts
chain8_endyr	integer	Year the chain8 history ends
chain8_ipcc_system	char(39)	IPCC land use system for the chain8 history
chain8_ipcc_grasslandother	char(9)	IPCC grassland/other class for the chain8 history
chain8_ipcc_mgmt	char(20)	IPCC management class for the chain8 history
chain8_ipcc_input	char(19)	IPCC Carbon inputs class for the chain8 history
Id	integer	primary key field
Datetime	timestamp(14)	timestamp field

Preparing Input Data for Simulation Runs

The datasets necessary for the model runs can all be prepared from the "Generate Files" page of the GUI, as described in *Appendix 2: Tutorial on using the Graphical User Interface*. The data should be generated in the following order:

1. Crops data (/home/gefsoc/model_runs/crops.dat)
2. Histories data (/home/gefsoc/model_runs/histories.dat)
3. Run the checkblock routine to test for problems with the crop histories.
4. Soils data ((/home/gefsoc/model_runs/soils.dat)
5. Climate data (/home/gefsoc/model_runs/gefweather.wth).
6. Run table/file (/home/gefsoc/model_runs/gefregion.dat)
7. Management Sequences ((/home/gefsoc/model_runs/gefexp.dat)
8. Soil Area Proportion ((/home/gefsoc/model_runs/gefruns.dat)
9. Replace the output data table with a new one corresponding to the regression intervals appropriate for this experiment.

After completing these steps, one should proceed to running the PERL scripts that run the models and generate the model output, as described in the next section.

Example Dataset

Following is an example of a hypothetical management system specification. The example is taken from the Northern Great Plains of North America, in an area that contains two distinct land use types- forested, mountainous country primarily managed for timber and rangeland grazing, and level plains primarily used for annual grain and hay crops as well as rangeland grazing.

Base Polygon: Major Land Resource Area (MLRA) 52.

Equilibrium Period: 7000 years
Equilibrium Vegetation Type: mixed grass prairie
Equilibrium Crop.100 Type: 75% C3 grasses, 25% C4

Historical Land Management: Dryland annual grain production in upland soils, irrigated hay production in lowland soils, and dryland rangeland grazing in upland soils.

Base Period: 1880-1974

- Base #1: annual grain production
    - 1880-1900: continuous spring wheat (no fertilization, very low production wheat variety, heavy tillage), grazing (heavy)
    - 1901-1920: fallow-spring wheat (no fertilization, low production wheat variety, heavy tillage), grazing (heavy)
    - 1921-1950: fallow-spring wheat (low fertilization, moderately production wheat variety, heavy tillage), grazing (heavy)

    - 1951-1974: fallow-spring wheat (moderate fertilization, high production wheat variety, moderate tillage), grazing (heavy)
- Base #2: lowland hay production
    - 1880-1974: continuous grass hay, no fertilization, flood irrigation.
- Base #3: rangeland grazing
    - 1880-1950: continuous heavy grazing, no amendments.
    - 1951-1974: continuous moderate grazing, no amendments.
- Base #4: native pine forest/grassland
    - 1880-1974: continuous light grazing, fire suppression

Recent Period: 1975-1994

- Annual grain cropping systems: grassland reserve (no tillage, no fertilization, no grazing), fallow-spring wheat (moderate fertilization, high production wheat variety, moderate tillage).
- Irrigated hay cropping systems: continuous grass hay, no fertilization, flood irrigation.
- Rangeland grazing: continuous heavy grazing, continuous moderate grazing.
- Forest land: native forest, clearcut forest, partial cut forest, regenerating forest.

Current Period: 1995-2004

- Annual grain cropping systems: grassland reserve (no tillage, no fertilization, no grazing), fallow-spring wheat (moderate fertilization, high production wheat variety, moderate tillage), and fallow-spring wheat (moderate fertilization, high production wheat variety, no tillage).
- Irrigated hay cropping systems: continuous grass hay, no fertilization, flood irrigation.
- Rangeland grazing: continuous heavy grazing, continuous moderate grazing, and intensive rotational grazing.
- Forest land: native forest, clearcut forest, partial cut forest, regenerating forest.

Future Period: 2005-2030

- Annual grain cropping systems: grassland reserve (no tillage, no fertilization, no grazing), fallow-spring wheat (moderate fertilization, high production wheat variety, moderate tillage), and fallow-spring wheat (moderate fertilization, high production wheat variety, no tillage).
- Irrigated hay cropping systems: continuous grass hay, no fertilization, flood irrigation.
- Rangeland grazing: continuous heavy grazing, continuous moderate grazing, and intensive rotational grazing.
- Forest land: native forest, clearcut forest, partial cut forest, regenerating forest.

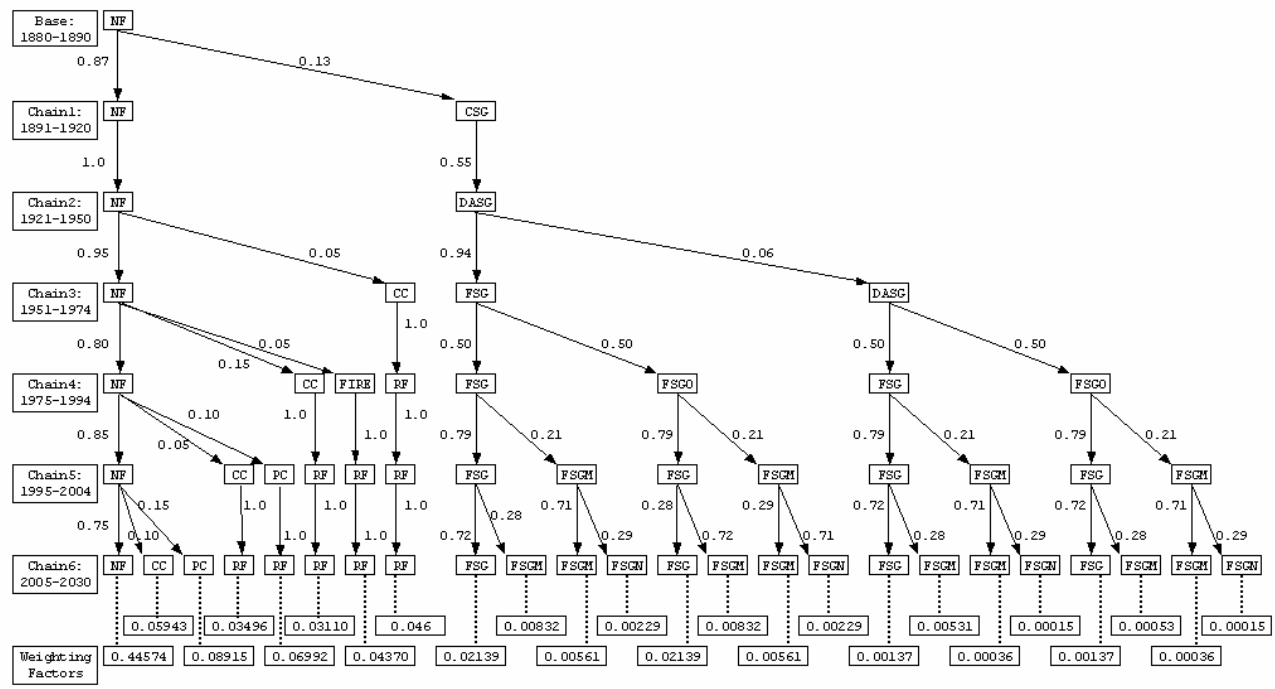


Figure 13. Management sequence diagrams for MLRA 46. Management system abbreviations are as follows: NF = native forest, CC = clearcut tree removal, PC = partial cut tree removal, FIRE = stand-replacing fire, RF = regenerating forest, CSG = continuous small grains, DASG = dryland alfalfa-small grain, FSG = fallow-small grain (conventional tillage), FSGO = fallow-small grain-oilseed, FSGM = fallow-small grain (minimum tillage), FSGN = fallow-small grain (no tillage).

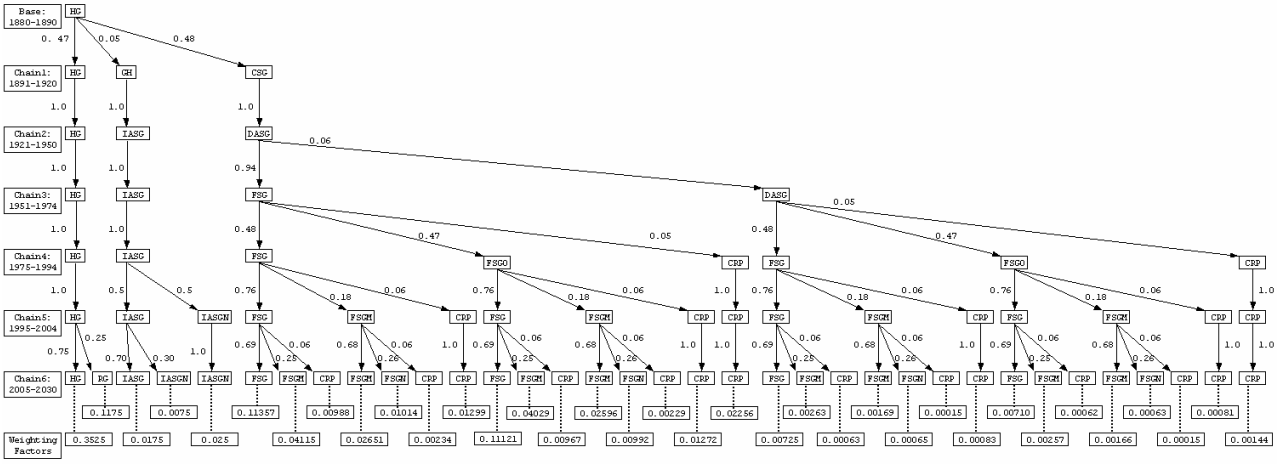


Figure 14. Management sequence diagrams for MLRA 52. System abbreviations are as follows: HG = heavy grazing, GH = grassland hay, IASG = irrigated alfalfa-small grain (conventional tillage), IASGN = irrigated alfalfa-small grain (no tillage), RG = rotational grazing, CSG = continuous small grains, DASG = dryland alfalfa-small grain, FSG = fallow-small grain (conventional tillage), FSGO = fallow-small grain-oilseed, FSGM =

fallow-small grain (minimum tillage), FSGN = fallow-small grain (no tillage), CRP = Conservation Reserve Program.

Table 7. Summary of management sequences for MLRA 46.

Management Sequence ID	Base	Chain1	Chain2	Chain3	Chain4	Chain5	Chain6	Weighting Factor
NF-NF-NF-NF-NF-NF-NF	NF	NF	NF	NF	NF	NF	NF	0.44574
NF-NF-NF-NF-NF-NF-CC	NF	NF	NF	NF	NF	NF	CC	0.05943
NF-NF-NF-NF-NF-NF-PC	NF	NF	NF	NF	NF	NF	PC	0.08915
NF-NF-NF-NF-NF-NF-RF	NF	NF	NF	NF	NF	NF	RF	0.03496
NF-NF-NF-NF-NF-CC-RF	NF	NF	NF	NF	NF	CC	RF	0.06992
NF-NF-NF-NF-CC-RF-RF	NF	NF	NF	NF	CC	RF	RF	0.13110
NF-NF-NF-CC-RF-RF-RF	NF	NF	NF	CC	RF	RF	RF	0.04370
NF-NF-NF-CC-RF-RF-RF	NF	NF	NF	CC	RF	RF	RF	0.04600
NF-CSG-DASG-FSG-FSG-FSG-FSG	NF	CSG	DASG	FSG	FSG	FSG	FSG	0.02139
NF-CSG-DASG-FSG-FSG-FSG-FSGM	NF	CSG	DASG	FSG	FSG	FSG	FSGM	0.00832
NF-CSG-DASG-FSG-FSG-FSGM-FSGM	NF	CSG	DASG	FSG	FSG	FSGM	FSGM	0.00561
NF-CSG-DASG-FSG-FSG-FSGM-FSGN	NF	CSG	DASG	FSG	FSG	FSGM	FSGN	0.00229
NF-CSG-DASG-FSG-FSGO-FSG-FSG	NF	CSG	DASG	FSG	FSGO	FSG	FSG	0.02139
NF-CSG-DASG-FSG-FSGO-FSG-FSGM	NF	CSG	DASG	FSG	FSGO	FSG	FSGM	0.00832
NF-CSG-DASG-FSG-FSGO-FSGM-FSGM	NF	CSG	DASG	FSG	FSGO	FSGM	FSGM	0.00561
NF-CSG-DASG-FSG-FSGO-FSGM-FSGN	NF	CSG	DASG	FSG	FSGO	FSGM	FSGN	0.00229
NF-CSG-DASG-DASG-FSG-FSG-FSG	NF	CSG	DASG	DASG	FSG	FSG	FSG	0.00137
NF-CSG-DASG-DASG-FSG-FSG-FSGM	NF	CSG	DASG	DASG	FSG	FSG	FSGM	0.00053
NF-CSG-DASG-DASG-FSG-FSGM-FSGM	NF	CSG	DASG	DASG	FSG	FSGM	FSGM	0.00036
NF-CSG-DASG-DASG-FSG-FSGM-FSGN	NF	CSG	DASG	DASG	FSG	FSGM	FSGN	0.00015
NF-CSG-DASG-DASG-FSGO-FSG-FSG	NF	CSG	DASG	DASG	FSGO	FSG	FSG	0.00137
NF-CSG-DASG-DASG-FSGO-FSG-FSGM	NF	CSG	DASG	DASG	FSGO	FSG	FSGM	0.00053
NF-CSG-DASG-DASG-FSGO-FSGM-FSGM	NF	CSG	DASG	DASG	FSGO	FSGM	FSGM	0.00036
NF-CSG-DASG-DASG-FSGO-FSGM-FSGN	NF	CSG	DASG	DASG	FSGO	FSGM	FSGN	0.00015

Table 8. Summary of management sequences for MLRA 52.

Management Sequence ID	Base	Chain1	Chain2	Chain3	Chain4	Chain5	Chain6	Weighting Factor
HG-HG-HG-HG-HG-HG-HG	HG	HG	HG	HG	HG	HG	HG	0.35250
HG-HG-HG-HG-HG-HG-RG	HG	HG	HG	HG	HG	HG	RG	0.11750
HG-GH-IASG-IASG-IASG-IASG-IASG	HG	GH	IASG	IASG	IASG	IASG	IASG	0.01750
HG-GH-IASG-IASG-IASG-IASG-IASGN	HG	GH	IASG	IASG	IASG	IASG	IASGN	0.00750
HG-GH-IASG-IASG-IASG-IASGN-IASGN	HG	GH	IASG	IASG	IASG	IASGN	IASGN	0.02500
HG-CSG-DASG-FSG-FSG-FSG-FSG	HG	CSG	DASG	FSG	FSG	FSG	FSG	0.11357
HG-CSG-DASG-FSG-FSG-FSG-FSGM	HG	CSG	DASG	FSG	FSG	FSG	FSGM	0.04115
HG-CSG-DASG-FSG-FSG-FSG-CRP	HG	CSG	DASG	FSG	FSG	FSG	CRP	0.00988
HG-CSG-DASG-FSG-FSG-FSGM-FSGM	HG	CSG	DASG	FSG	FSG	FSGM	FSGM	0.02651
HG-CSG-DASG-FSG-FSG-FSGM-FSGN	HG	CSG	DASG	FSG	FSG	FSGM	FSGN	0.01014
HG-CSG-DASG-FSG-FSG-FSGM-CRP	HG	CSG	DASG	FSG	FSG	FSGM	CRP	0.00234

HG-CSG-DASG-FSG-FSG-CRP-CRP	HG	CSG	DASG	FSG	FSG	CRP	CRP	0.01299
HG-CSG-DASG-FSG-FSGO-FSG-FSG	HG	CSG	DASG	FSG	FSGO	FSG	FSG	0.11121
HG-CSG-DASG-FSG-FSGO-FSG-FSGM	HG	CSG	DASG	FSG	FSGO	FSG	FSGM	0.04029
HG-CSG-DASG-FSG-FSGO-FSG-CRP	HG	CSG	DASG	FSG	FSGO	FSG	CRP	0.00967
HG-CSG-DASG-FSG-FSGO-FSGM-FSGM	HG	CSG	DASG	FSG	FSGO	FSGM	FSGM	0.02596
HG-CSG-DASG-FSG-FSGO-FSGM-FSGN	HG	CSG	DASG	FSG	FSGO	FSGM	FSGN	0.00992
HG-CSG-DASG-FSG-FSGO-FSGM-CRP	HG	CSG	DASG	FSG	FSGO	FSGM	CRP	0.00229
HG-CSG-DASG-FSG-FSGO-CRP-CRP	HG	CSG	DASG	FSG	FSGO	CRP	CRP	0.01272
HG-CSG-DASG-FSG-CRP-CRP-CRP	HG	CSG	DASG	FSG	CRP	CRP	CRP	0.02256
HG-CSG-DASG-DASG-FSG-FSG-FSG	HG	CSG	DASG	DASG	FSG	FSG	FSG	0.00725
HG-CSG-DASG-DASG-FSG-FSG-FSGM	HG	CSG	DASG	DASG	FSG	FSG	FSGM	0.00263
HG-CSG-DASG-DASG-FSG-FSG-CRP	HG	CSG	DASG	DASG	FSG	FSG	CRP	0.00063
HG-CSG-DASG-DASG-FSG-FSGM-FSGM	HG	CSG	DASG	DASG	FSG	FSGM	FSGM	0.00169
HG-CSG-DASG-DASG-FSG-FSGM-FSGN	HG	CSG	DASG	DASG	FSG	FSGM	FSGN	0.00065
HG-CSG-DASG-DASG-FSG-FSGM-CRP	HG	CSG	DASG	DASG	FSG	FSGM	CRP	0.00015
HG-CSG-DASG-DASG-FSG-CRP-CRP	HG	CSG	DASG	DASG	FSG	CRP	CRP	0.00083
HG-CSG-DASG-DASG-FSGO-FSG-FSG	HG	CSG	DASG	DASG	FSGO	FSG	FSG	0.00710
HG-CSG-DASG-DASG-FSGO-FSG-FSGM	HG	CSG	DASG	DASG	FSGO	FSG	FSGM	0.00257
HG-CSG-DASG-DASG-FSGO-FSG-CRP	HG	CSG	DASG	DASG	FSGO	FSG	CRP	0.00062
HG-CSG-DASG-DASG-FSGO-FSGM-FSGM	HG	CSG	DASG	DASG	FSGO	FSGM	FSGM	0.00166
HG-CSG-DASG-DASG-FSGO-FSGM-FSGN	HG	CSG	DASG	DASG	FSGO	FSGM	FSGN	0.00063
HG-CSG-DASG-DASG-FSGO-FSGM-CRP	HG	CSG	DASG	DASG	FSGO	FSGM	CRP	0.00015
HG-CSG-DASG-DASG-FSGO-CRP-CRP	HG	CSG	DASG	DASG	FSGO	CRP	CRP	0.00081
HG-CSG-DASG-DASG-CRP-CRP-CRP	HG	CSG	DASG	DASG	CRP	CRP	CRP	0.00144

Managing Regional Model Runs with the NREL Regional Century Scripting System

Introduction

The NREL Regional Century Scripting System is a set of tools for running large numbers of related Century runs. High fidelity simulations of soil carbon stocks and management changes at a landscape or regional scale requires dealing with a large number of environmental and management options and a variety of crops, management practices. The size of such a simulation often grows in a combinatorial fashion and can easily exceed several million individual model runs. The scripting system is designed to martial available computing resources to accomplish these simulations in an efficient and reliable fashion.

Overview

Calculation throughput set practical limitations to the size and complexity of detailed simulations. Here a regional simulation is defined as the sum of all model runs that simulate land use change on each unique intersection of climate region and base land use polygon. Hence, each simulation run can take up to several hours, depending on the number of soils and the complexity of the land use changes that are being simulated.

In Century, these histories run with weather and coil conditions do not interact. Organizing calculations along ecological boundaries and running different regions on separate hosts provides a natural strategy to decrease the calculation time by parallelizing the calculations. Organizing the calculations in this fashion and implementing this as a formal part of the system allowed us to take advantage of unused CPU cycles on existing workstations, and maximize throughput with a minimum of recoding, The parallelization was accomplished using a client/server model using inter-process communication to coordinate the calculations. This has proven to be an effective strategy to manage these simulations on single machines, loosely coupled UNIX workstations, and LINUX clusters like the NREL calculation server, a Beowulf cluster composed of 13, dual-CPU machines.

Since the system architecture recommended for the GEFSOC system consists of a single LINUX server node, a client/server model and its impact on the user interface run may seem superfluous. The parallel processing architecture is a major capability of the GEFSOC scripting system. The parallel processing architecture was retained for a number of practical reasons. The primary reason for that is that retaining the existing architecture allows for efficient utilization of the NREL hardware. This also makes the GEFSOC system extensible allowing more processing power to be brought on line as the problems get more complex without changing the user interface.

There are two types of simulation runs that have to be executed in this system- a set of *equilibrium runs* that initiate the model to an equilibrium state prior to the base histories, and a set of *land use model runs* that simulate the land use in the region to be modeled. The equilibrium runs are accomplished first in order to establish equilibrium soil conditions for the start of the base histories.

System Architecture

The regional simulation is organized into parallel history trees rooted at different environmental locations. These trees, or runs, can be executed with a minimum of interaction between runs if they are properly organized and isolated. These runs can then be allocated on a location bases as independent jobs on separate processes. Different processes can execute these runs without impacting the accuracy of the overall simulation.

This architecture takes advantage of this organization by allocating individual runs to calculation processes. Preventing run duplication and balancing the load to each of these client processes running on different hardware responding to different loads has proven has proven difficult. Experience has shown that a master process must take charge of the simulation and allocate the runs. The current architecture provides two functional elements to run a job - a model run *server* to manage the simulation, and one or more *calculation clients* to run the simulations.

The server reads in the text files that define the management region, divides up the work into a series of *simulation runs* (not to be confused with individual Century *model runs*), and then parcels out the jobs to the calculation client. It is also the users primary interface to manage the model runs. In the minimal hardware system specified, both the both the server and calculation client processes run on the LINUX server specified for this system.

Data coordination is handled by a combination of private working directories and file locking on shared data directories. The disk sharing tools available with current operating systems provide efficient and reliable data sharing between machines. Experience has shown that file locking techniques on shared volumes can handle data isolation and output problems inherent in the independent runs.

The Century and RothC programs were not modified to handle independent parallel calculations. To prevent runs sharing or overwriting working files, each calculation process provides a local working directory. This working directory is normally on local disk storage to reduce network traffic and to reduce the load on the disk sharing processes to the critical data sharing requirements.

To run calculation clients on multiple machines, the simulation description files must be placed on a network drive. Most of simulation description is read during set up and retained locally by the calculation clients. The server process alone has access to management files with the control information passed by inter-process communication.

The MySQL server handles most output data collection and the associated files. The exception to that are the equilibrium archives. These binary files contain information needed to restart the models and are common to the clients. Currently, clients buffer these archives and the system writes the results to the network drive using a file locking process or the server to prevent collisions. The default locking algorithm provided with the GEFSOC system is expected to work with most industry standard network file sharing protocols but operating system parameters can cause problems. In such cases the server can mediate the file writes.

Management

This multi process architecture is present even if only one calculation machine is used. This run organization has several practical consequences for managing a simulation. Practically this means that the manager must be aware of the multiple process running and manage the simulation through the server interface. Also this means that error handling and debugging is different; allowing more error recovery options at the price of increased complexity.

A primary concern is the number of processes to devote to a simulation. Even with a single processor machine, it is possible to run multiple servers and calculation clients. It is never beneficial to run multiple simulation servers for any simulation. The user should watch the simulation logs and stop any extraneous servers.

Testing has proven that the calculation processes are highly CPU limited. A good rule of thumb is that a machine can run one calculation process CPU as long as all processes fit in RAM. Beyond that limit, additional clients may either slightly increase or significantly decrease throughput. If the RAM cannot contain more than one client, the additional swapping IO will significantly reduce throughput. It is also possible to allocate to many clients for a simulation. Adding clients increases the load on the common input/output channels. This increasing load will eventually overload the MySQL and disk servers and strangle throughput. This can lead to long and variable IO times and can lead to client termination due to network errors.

The Run Server

A small, independent, process called a run server controls the progress of a simulation. Its purpose is to tell the calculation clients which piece of the simulation to work on and to provide information on the status of the simulation. It does none of the calculations or data manipulation, but it is essential to coordinate and control the progress of the simulation. A simulation can be monitored, managed, and partially debugged by understanding and controlling the server.

The server goes through five steps to manage a simulation. These five steps are startup, opening a communication channel, advertising its presence, managing the clients and finally shutdown. The command to start a server can be given by the user or by a calculation process. This command creates a small independent process resident on the computer. Like any good manager, the server's first task is to open lines of communication. It is a fatal error if this communication channel, or port, is not available. Once the process is running with an open port, it then creates a small status file, normally called GefRuns.run, to advertise the server address and display the simulation status. The server spends most of its time taking messages and answering questions that mange the run. By sending the server commands, the user can change the server state and thus manage the simulation. The last task done by the server is to summarize the run and shutdown.

Starting the Server

The initial step in running any simulation is to start the server. This can be done either by the user or by a calculation process. The server start command creates a small independent process resident in the computer. server itself is a small independent process.

The server can be started in one of two ways. The easiest way to start the server is let a calculation client spawn the required server. However it links the server and the calculation process and certain fatal errors or kill commands can stop both processes leaving the simulation in an inconsistent state.

The safest way to start a simulation is to use a program called startUDPserver.pl.
The command is:

/usr/local/nrel/bin/startUDPserver.pl <-r| -e| -s> runfil.dat

where

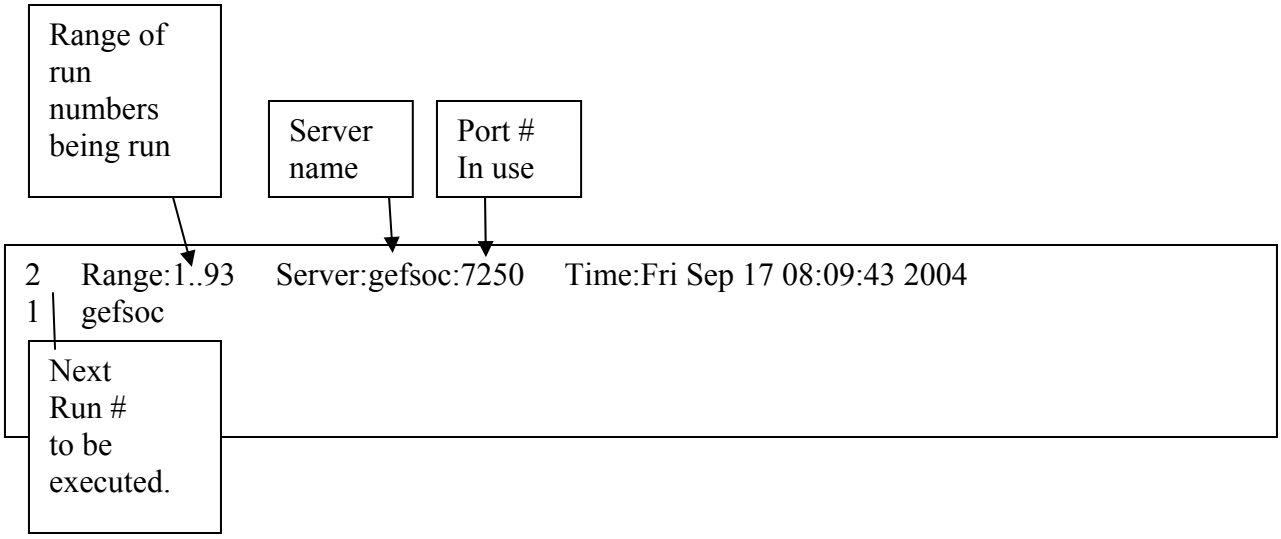
-s run#	starts execution at run # from the file /home/gefsoc/model_runs/gefsoc1/gefruns.dat
-e run#	stops execution at run # from the file /home/gefsoc/model_runs/gefsoc1/gefruns.dat
-r'r1,r5..r16'	executes a list of one or more run numbers from the file /home/gefsoc/model_runs/gefsoc1/gefruns.dat. Runs are specified by a comma delimited list enclosed in quotes. Consecutive run numbers can be abbreviated by the initial and final number separated by by periods, 5,6,7,8 becomes 5..8.
<runfil.dat>	the soil run table to allocate runs from -

Although running startUDPserver.pl before initiating the model runs requires an additional step, it provides a server that is independent of any calculation client so that if the calculation client process fails, the server will not be affected.

Server status

The server creates a file that it uses to display the run status. The file has a default name taken from the simulation soil run table with the extension changed to '.run'. As an example, if the soil run table is called gefruns.dat then the status file will be gefruns.run. The server presents this file to identify itself to any clients as they join the execution queue. It also is the primary method used to tell the execution status.

This status file is a copy of a data structure internal to the server. Modifying the file does not effect the simulation. In fact, the status file can be deleted without affecting the progress of a simulation. However, without the file, new calculation clients do not know a server is present and thus will try to restart the calculation.
An example of the file format looks like this:


Range of run numbers being run
Server name
Port # In use
2 Range:1..93 Server:gefsoc:7250 Time:Fri Sep 17 08:09:43 2004
1 gefsoc
Next Run # to be executed.

The first line of that file presents the execution status. It presents:

1. The next run execute,
2. Range followed by the runs to execute,
3. Server followed by the server name and port,
4. Time followed by the simulation start time.

The remaining lines give the an active run number, the process name currently running that run and the time the run was checked out. The list of processes is presented in chronological order with the run taken most recently on top. Processes on slow machines, those that are not getting system resources fall to the bottom of the list. Jobs that have failed stay on the bottom of the list and readily apparent from the start times.

Communicating with the Server

The proper way to manage a simulation is to control the server. By issuing the proper commands, the server can stop the simulation, rerun a failed soil, run a subset of the runs, or remove a client from the execution cue. Any subset of a simulation can be described with the –s, -e, and –r switches. Once the simulation starts, there are a limited number of commands ta
The utility srvrcmd is provided to communicate with the server. This program sends commands to the addressed server. It then waits for any response from the server and prints the response to the terminal. The proper syntax is:

/usr/local/nrel/bin/srvrcmd ***machine:port command***

where

machine:port is the server's IP address or DNS name and the port number that is being used. The address of the server, ***machine:port***, is given in both the server log and the status file.

command one or more words to be sent to the server (see below).

The server address, ***machine:port***, can be abbreviated or omitted and srvrcmd will search for a server according to the following rules:

- If the server name is given the program will search for a server on the specified machine in the through the allowed ports,7250..7258.
- If the server name is omitted, it will first open any status file, <.run>, in the working directory will be opened and the message will be sent to the listed server.
- if there is no status file the program will search the allowed ports on the users (local) machine and any other machines allowed for in the user defined server list.

The scan process provides a report on any address it tests.

The server recognizes a fairly limited set of commands. For security reasons, any command not recognized is responded to with a 0, but no action is taken. The following management commands are recognized:

Command	**Response**
hello	This is a non destructive command that verifies the server is running. The returned values are the current run number, the status file (the .run file), and the run list
abort	immediately stops the server and deletes the status file
shutdown	stops all clients at the end of the current run and shuts down the server when all clients finish
restart <process>	mark the run done by <process> for rerun. This is an error recovery command.
repeat r#	repeat a completed soil run number. This is an error recovery command.
sayagain	repeat the response from the previous command.
logout <process>	mark <process> to be shut down at the end of the current run without effecting other client process.

There are also a number of programming commands that are used by the scripts that are used by the scripts to request control information and are not needed for normal management functions. Stopping the server

The server will delete its status file and die gracefully after the clients have completed all the runs identified in the input files. The proper way to stop a simulation is to issue the “logout” command.

The server will then shutdown any clients and itself. A dormant or orphaned server, can be stopped with the abort command.

Debugging simulation and server problems

The server is a simple program so most server problems indicate a problem with the underlying simulation. The server specific errors show up during server startup. The first of these problems have to do with parsing the input command. A series of messages ending with “ERROR: Unknown argument” means that the input command was not correctly formatted or a bad switch was input. One common cause of this error is the run switch, -r, followed by a list of runs that are NOT enclosed in quotes. If the list is not enclosed in quotes, the shell parses the comma characters and mangles the list. Another command error is, “ERROR: Missing run control file gefruns.dat”. This means that the program was unable to find the run file. Check the spelling, including capitalization, of the file name and make sure it is in the current working directory or the path is correct.

One serious server startup error is ERROR: no UDP ports available". This happens when there are no communications channels, or ports available in the server’s range of 7250-7258. Since these ports are not normally in use, the probable cause of this is the presence of a number of forgotten servers. The solution is to abort orphaned servers until srvrcmd reports “no server responded”. If the simulation doesn’t then run consult your system administrator to help determine available ports.

The remaining problems reflect coordination problems between the server and the simulation. There are three types of server problem, a server with no clients, clients with no server, or a server status file without a server. Other than the occasional operator error, these conditions occur when errors, or user aborts, cause clients, or the server, to die.

The most common of these problems is a dormant server; one that has no clients working. Dormant servers normally have an associated status file, GefRuns.dat. Assuming that the calculation clients were started, this happens when the clients die without completing one or more runs. The uncompleted soil runs are listed in the status file along with the process alias that was working that run. The last part of the process log should be carefully reviewed to make sure the client is dead and not simply working on a long run. If the client is dead, there should be error messages designed to identify the fatal condition and to assist the user correcting the problem.

Dormant servers will interfere with an attempt to restart the simulation since the start process checks to see if the simulation is running. This is done to ensure that multiple simulations are not running. Further, to assist debugging, the identification of the failed run is not automatically cleared to ensure that the condition that terminated the soil run is corrected before the run is attempted again. Once the fatal condition is corrected, finishing the simulation can be done by two methods. Often the simplest is to restart the simulation by first removing the old results from the MySQL tables, and then issuing the abort command to the existing server and then restarting the simulation.

A long run can be completed with a similar process. Again the bad input must be corrected. This is complicated by the fact that is is impossible to change the input to a running client. Thus any running clients must be stopped either by user intervention or through normal completion. The input files can be then be updated. Any dead runs can be marked for restart and the calculation clients restarted.

An orphaned server is one that has lost its status file. If there are enough of these orphaned servers then new simulations will fail do to a lack of communication ports. Orphaned servers can be dealt with in two ways. A hello command will cause the server to rewrite its status file. Then the simulation can be restarted or terminated as before. The server can be aborted directly. To prevent servers from being orphaned, run status files should NOT be deleted unless it is certain that the associated server has been terminated.

The proper way to stop a server is to issue the 'srvrcmd abort' command. If multiple servers are running, the command can be issued several times until srvrcmd reports “no server responded”.

The final condition is a status file with no server. This file advertises the presence of an active simulation. If a server is started, the new server will attempt an error recovery and try to finish the interrupted simulation. This generates a significant startup message, “Warning: restarting the server”, that should not be ignored. Strictly speaking this is not an error but an error recovery message. However, if you get this message and you were NOT expecting to restart a simulation, you should abort the server, make sure the status file has been removed then restart the simulation.

Starting the calculation clients

The calculation clients are independant scripts that actually run the equilibriums or the management sequences.

The clients can be started from a terminal window either directly on the LINUX server or via a SSH terminal window from the PC to the LINUX server. The calculation clients should be submitted as cron (batch) jobs, or remote procedure calls so the jobs are not tied to the user’s terminal. For example, the commands should look like:

```
cd /home/gefsoc/model_runs/gefsoc1
perl ./gefeq.pl > ./jobrun1log
```

Alternatively, a user can run the commands like this:

```
rsh -n 10.10.11.101 'cd /home/gefsoc/model_runs/gefsoc1/; nice gefeq.pl > ./jobeq1log
rsh -n 10.10.11.101 'cd /home/gefsoc/model_runs/gefsoc1/; nice gefrun.pl > ./jobrun2log
```

OR

```
at -m now cd /home/gefsoc/model_runs/gefsoc1/ ; nice gefeq.pl > ./jobeq1log
at -m now cd /home/gefsoc/model_runs/gefsoc1/ ; nice gefrun.pl > ./jobrun2log
```

A complete explanation of these commands is left to a LINUX user’s manual but basically they are designed to do four things:

1) First the ‘cd’ command ensures the working directory is the main directory of the analysis (the ‘cd’ command);
2) they lower the job's priority so that the user can still work (the ‘nice’ command);
3) they spawn a separate process from the terminal window (the ‘rsh’ or ‘at’ command), so that the terminal window is not locked up by the process until it finishes; and
4) they redirect standard output and standard error to a log file.

These scripts generate a lot of output- up to several dozen lines for each unique combination of soil, climate, and management sequence. Since they are so verbose, the output should be redirected to a text file for all but the shortest of test runs.

The calculation client spawns a temporary file (/tmp/Centx/Cscript1) that it uses to store process information about the simulation run it is executing. The user can monitor the process of the model runs on the individual calculation client by reading the contents of this file.

Stopping the calculation clients

The best way to stop a client is to "logout" the process or "shutdown" the server. This allows the clients to save their data and gracefully clean up after themselves.

Calculation clients can be stopped in a faster and messier fashion by logging into the linux machine and issuing a kill command to the process id listed in the job's. This leaves the temp directory, again listed in the log file, with all of the working files. The temp directory, located in /tmp/centx or /scratch/centx, can be used for diagnostics, can be removed, or simply be ignored. If the directory is ignored, later simulations will remove the abandoned working directories.

Debugging the calculation clients

This is a complex and difficult task. Most errors identify installation problems, bad commands, missing or poorly constructed data files, or inconsistent problem definition in the data files. A second class of errors is from bad data input to the system. Bad input may produce explicit error messages but often can only be detected by critical examination of the results. It is important to remember that the system produces NO “routine” error messages. Any flagged error must be identified and corrected before data analysis can proceed.

Each client should have had its terminal output written to its own log file. These log files are the primary tool for locating and identifying errors.. To assist the user in locating errors a tool, *testlog*, is provided to scan log files looking for known error classes. It is important that users run testlog on ALL log files after each regional model run is complete, to make sure the runs are free from known errors before data analysis proceeds.

Examples of running the testlog tool:

```
cd /home/gefsoc/model_runs/gefsoc1/;
./testlog jobeq1log;
```

OR
./testlog /home/gefsoc/model_runs/gefsoc1/*log;

The first example will report errors from only one file (jobeq1log) in the /home/gefsoc/model_runs/gefsoc1 directory. The second example will report errors from all files in the directory that end with the letters “log”.

There are three classes of errors that testlog identifies.

Century Errors or gefeq.pl/gefrun.pl trapped errors

The first class are the fatal errors identified by Century or the run scripts. These are identified by the words "ERROR:" followed by an informative message. The system scans Century output and will copy the terminal output of any abnormal run to the log file. Century runs with fatal errors end with the message "abnormal termination". The severe "ERROR:" class messages are normally fatal and lead to the termination of the client. There is a large number of fatal errors. The philosophy here is that a long simulation with bad input is wasted time at best or at worst will lead to a proliferation of erros. It is better to identify problems early, stop the simulation, and allow the user to correct the input. Naturally, conditions that terminate Century, RothC, or one of the control scripts must be identified and corrected before the simulation will complete.

A similar class of errors is identified as warnings. Warning messages identify conditions that are abnormal but recoverable. Again, warnings are not considered normal and must be understood by the user at least to ensure that the recovery actions are appropriate to the current simulation. More sever warnings can terminate a section of the calculation but leave the process running. This is a can be done by core functions that are used where an immediate shutdown is not acceptable. These warnings may leave values designed to produce unusual condition so the unusual conditions are later identifiable.

PERL errors

A second class of error messages is in the PERL format. PERL normally identified its errors with a line and the package where the error occurs. These errors can be data errors trapped by internal routines or errors like undefined variables, data type mismatches, numeric operations on non-numeric data, or other PERL variable errors. These PERL message errors may not be fatal. If not fatal, these errors are often flagged multiple times since the offending data may be used in other locations or the same error is encountered on later iterations. These errors normally indicate that the input datafiles are not identified correctly, or are poorly constructed and should not be considered normal. Errors in the output routines means that the Century output is not matching the expected formats and normally means the inputs are not consistent.

IEEE errors

The third problem class is IEEE errors that come from the FORTRAN and C programs that execute the Models. These usually indicate that input data is improperly formatted or that the required

values are outside normal conditions. For example, clay values of 200% or maximum temperatures of 85 C are perfectly proper floating point numbers but they describe aphysical environmental values. Such values can cause the models to do strange thing things and generate IEEE errors like underflows, overflows or divide by zero errors. This type of error can come about from bad input data or, as a result of a improperly formatted input, reading the wrong value for the clay input. The errors can be either poor values in the Century .100 input files or bad data in the input tables that the system uses to describe the region. The inputs for all segments in the history chain must be checked since unusual conditions can propagate forward in time leading to errors as the simulation diverges from the proper solution.

Reading a calculation log file

To execute a run, a calculation client must go through a number of stages. First it must contact the server, or if one is not present it creates one. It then parses any input flags and creates its own working or temp directory. The input data is parsed and the output is created. Then it starts the calculations. These stages are identifiable by the messages in the log file. Each section gives information on the status of specific parts of the run process.

The first task is to contact the run server. If the server has been started in advance, the client will tell you which server is running.

open IOsocket 127.0.0.1 port 7250
Opened existing UDP Server:127.0.0.1:7250 for file GefRuns.run
the job label for pid 23933 is “gefsoc.'

If the attempt to contact a server fails, it will try and start a server and the first part of the log will show the server process being forked. The server gives a summary of the server job including some information on the server's health. Take a look at the server address. Increasing port numbers indicate lost servers that have not properly shut down

UDP server: gefsoc.local. port: 7250
runf: gefruns.dat
statf: gefruns.run
Runs: 1..10 forked UDP server at Tue Sep 28 19:09:00 2004 pid: 15302
the job label for pid 23933 is “gefsoc.'

This section ends with the process identifying its process id and its working name.

To prevent Century and RothC runs from interacting, a client sets up its own working directory. The process logs the default and working directories and what files it is copying to the working directory. Since there are probably several copies of the Century files available, make sure the job is using the files you expected.

The next section tells you about the run client, what data files it is reading, where the output is going and what runs the simulation is running. This information is NOT just a simple echo of the input parameters. The run scripts have the ability to search multiple directories for input files. A primary example of this are the Century .100 files. There are default copies of these file in /usr/local/nrel/LATEST.100/ however any simulation specific modifications should be placed in the working directory, gefsoc1.. Another common source of data mismatch occurs because linux is

case sensitive and DOS/Windows is often not. Thus gefregion.dat is NOT the same file as GefRegion.dat. Reviewing the input data echo can help eliminate most of these problems.

```
crop file  /home/gefsoc/model_runs/gefsoc1/crops.dat

copy  /Users/kendrick/WorkFiles/Gef/Jorden/GefRuns.esa     GefRuns.esa
reading climate from: /Users/kendrick/WorkFiles/Gef/Jorden/gefweather.wth
reading the region description from /Users/kendrick/WorkFiles/Gef/Jorden/gefregion.dat
<…>
Result File:   MySQL:althenia.nrel.colostate.edu:gefsoc_jordan_20050629:tbl_gef_output
Status File:   GefRuns.run
Run started on:  Fri Jul 29 15:42:47 2005
Running from  1  to  1
```

The next block is the status as it starts the run loop. The loop starts by making contact with the server. The GEFSOC system then identifies the management region and pulls up the management histories defined for the region. To start every run block, it then repeats the run information it receives from the server along with the area fraction under the management protocol. It follows with the labels for every schedule that is being prepared for the current history chain.

```
gefsoc  run  '1','SIL','N','05001','131','-',0.2    Time: Tue Sep 28 19:09:02 2004

==================================
BuildEvt:   255_N
```

It then goes into the execution phase of the run. Expect to see enough of the Century command echoed to identify the position in the history chain. At the end of the history chain it gives the elapsed timing information for the chain, looking something like this:

```
Centx -q Events/164_N -R1 -N 164_N
Centx -q Events/109 -e 164_N -R1 -N 109
Centx -q Events/168 -e 109 -R1 -N 168
Centx -q Events/118 -e 168 -R1 -N 118
Centx -q Events/156 -e 118 -R1 -N 156
Centx -q Events/157 -e 156 -R1 -t 157.lis -dt –v
elapsed time: 0:0:8  for 1 histories or  0.1 runs/sec
```

These run blocks continue until all runs are complete.
The end of the log file gives the shut down information. It saves all the event files from the last history chain to the Events subdirectory for debugging purposes. The final message is the total number of Century calls.

```
copy  /tmp/centx/Cscript0/Events/255_N.evt /home/gefsoc/model_runs/gefsoc1/Events/255_N.evt
copy  /tmp/centx/Cscript0/Events/262_ct.evt /home/gefsoc/model_runs/gefsoc1/Events/262_ct.evt
copy  /tmp/centx/Cscript0/Events/309_ct.evt /home/gefsoc/model_runs/gefsoc1/Events/309_ct.evt
saved eventfiles from  /tmp/centx/Cscript0/Events/
9 Century calls; removing working directory     Time: Tue Sep 28 19:09:05 2004
```

If the calculation client started the server (or if both are being run from the same terminal) there will be a shutdown message from the server.

```
SERVER shutdown  Tue Sep 28 19:09:05 2004  - deleting  GefRuns.run
```

Debugging Crop History problems

Running the Century Block Check routine

The modeling system includes a tool to help users find problems with Century events, files before model runs are attempted. The program checks for a number of common problems cultivation during the growing season or harvest, extended fallow, events with undefined arguments, duplicate events in a month, bad FERT and OMAD syntax, or missing data in the flat files.

This check is a PERL script located at

/usr/local/nrel/perl/checkblock1.pl.

This script imports the histories.dat and crops.dat ASCII flatfiles (which contain the exported contents of the land management database database) and builds schedule files from each of crop history, testing for problems that may arise during the actual model run.

To run the block check routine, use the following syntax:

```
cd /home/gefsoc/model_runs/gefsoc1/
/usr/local/nrel/perl/checkblock1.pl ./crops.dat ./histories.dat
```

This routine builds a Century event file for every crop history in the database. The event files named by appending the ‘evt’ extension to the history ID and then placing the event file in the Events subdirectory to the current working directory. For example, the schedule file defined by crop history id 64 will be ./Events/64.evt.
The format of the checkblock file is defined by two major sections. After the program starts checking the first history it echoes back the available options from the Century “.100” files. This looks like:

```
**************************************************************************
--- checking crop block @1975-1994 grazing moderate_57 -- First Growth month 3 -- length 1
tree   HRDWD CONIF CONOR PRTP  HRWD6 BFGN  HFR   CWT
irri   APET  A50   A25   A15   A75   A95   AF    F5    FLOOD AF95
crop   W0    W1    W2    W3    WW2S  WW3S  W4    SW1   XSW2  SW2   XSW3  SW3
     C-HI  C6    CDRY  C5    C5X   C4    C3    C2    C1    C     CDL   SORG
     M     XM    COT   PEA   SYBN  SYBN1 SYBN2 RICE  POT   SUGB  SUGBL DBEAN
```

SUN CAN FSORG OAT1 OAT2 OAT3 BAR1 BAR2 BAR3 CLV1 ALF2 ALF GCP1
GCP GGCP GGC1 EGCP TG E G5 GI5 G4 GI4 G3 GI3
G2 GI2 G1 GI1 JTOM JONI WC3 CANE CANE2 RICL TOB PNUT
RVEG
omad BFSD BFSL BFLQ DYSD DYSL DYLQ PYSD PYSL PYLQ SWSD SWSL SWLQ
HRSD HRSL HRLQ SHSD SHSL SHLQ MISL MILQ MLRA SW P H
SH C B IM80 IM60 IM40 IM30 IM20 IM10 M W M0.1T
M0.2T M.25T M0.3T M0.4T M0.5T M0.6T M0.7T M0.8T M0.9T M1.0T M1.2T M1.5T
M2.0T M2.5T M3.0T M4.0T M5.0T M5.5T M6.0T MPV
trem CLEAR BLOWD FIRE SAMP
cult A B C D E F G H I J K CTIL
TP NDRIL DRILL HERB SUMF SHRD P S CULT ROW R
fire C M H
graz GL GM GH GVER G GCS W P LRRL LRRM LRRH LRRX
fert A A90 A75 A80 MED MAX N40 N36 N33.6 N31 N28 N25
N23 N22.4 N21 N20 N18 N17 N14 N11 N10 N9 N8.4 N8.2
N8 N7.9 N7.5 N7 N6.6 N6 N5.6 N5 N4.5 N4 N3.5 N3
N2.5 N2.1 N2 N1.8 N1.5 N1 N0.9 N07 N0.6 N05 NTST N4P2
P2 PS1 PS2 PS3 PS4 PS5
harv G GS G75S G90S HS ROOT R H H76 HAY MOW T
SIL GR CANE

The program continues through every new blocks for each history. Since a block may be used in more than one history, it only prints the check the first time it is encountered. At a minimum, the program gives the block name, id, block repeat length and the month where growth starts. The checks look like:

--- checking crop block @1975-1994 irrigated continuous corn_62 -- First Growth month 5 -- length 1
growing plant cultivation H in month 6

--- checking crop block @Equilibrium grass_68 -- First Growth month 3 -- length 15

--- checking crop block @2005-2030 grassland reserve_65 -- First Growth month 3 -- length 1

--- checking crop block @1975-1994 fallow-spring wheat_52 -- First Growth month 17 -- length 2
NOTE: block starts with a 17 month fallow
Application Rate OMAD '(0.5*BFSD)'

--- checking crop block @2005-2030 grazing conservation_60 -- First Growth month 3 -- length 1

--- checking crop block @1995-2004 fallow-spring wheat_53 -- First Growth month 16 -- length 2
NOTE: block starts with a 16 month fallow
Application Rate OMAD '(0.3*BFSD)'

--- checking crop block @pre-1890 continuous spring wheat_48 -- First Growth month 5 -- length 1

--- checking crop block @1891-1920 continuous spring wheat_49 -- First Growth month 5 -- length 1

--- checking crop block @1921-1950 fallow-spring wheat_50 -- First Growth month 17 -- length 2
NOTE: block starts with a 17 month fallow
Application Rate OMAD '(0.5*BFSD)'

--- checking crop block @1951-1974 fallow-spring wheat_51 -- First Growth month 17 -- length 2
NOTE: block starts with a 17 month fallow
Application Rate OMAD '(0.5*BFSD)'
Application Rate OMAD '(0.5*BFSD)'
****

The complete list of errors, warnings, and debugging statements reported are as follows:

- **First Growth month –** Reported for every block, and can be used to flags a possible error where the growth month starts late in the growing season, possibly due to a late PLTM, TFST, or FRST event or problems with over wintering crops.
- **NOTE: block starts with a N month fallow –** This is a warning that the block starts with an extended fallow period. These should be checked to make sure that fallow is not a result of a specification error.
- **ERROR: bad event –** The event option indicated has no corresponding entry in the .100 file.
- **Duplicate I<EVENT> event in month –** There are two or more events of a type <EVENT> in the specified month. Although not a fatal error, only the second event will be modeled by Century.
- **Harvest cultivation in month –** There is a cultivation event in the same month of harvest. Century models cultivations before harvests, which is likely to divert material from the harvest into standing dead, litter, or soil C pools.
- **growing plant cultivation <type> in month –** There is a cultivation event of <type> that may harm or kill growing plants in the specified month. The warning can be ignored if the tillage event will not kill, or has a small effect, on growing plants. Consider moving the event to the month of planting, reducing its intensity.
- **WARNING: bad fertilizer syntax <OPT>–** The in schedule fertilizer amount is not parsable. User must revise the FERT event syntax specified.
- **Application Rate OMAD <wt> –** This is an option that weights the OMAD option by the specified weighting factor.
- **Immediate OMAD (replace ASTGC) <ASTGC> –** This is option replaces the C addition amount listed in omad.100, ASTGC, with the value listed .
- **WARNING: bad OMAD syntax -** The in schedule OMAD amount is not parsable. User must revise the OMAD event syntax specified.
- **Warning: bad history definition –** There is a missing rotation block in the history indicated. Check the rotations and history in the GUI and make sure that the correct rotations are specified.

- **WARNING: CropHistory skipping duplicate history key <ID>–** Input processing has detected two histories with the same ID. History ID's must be unique. Consider this a critical database error.
- **WARNING: skipping duplicate history name <ID> –** Internal Error. Input processing has detected two histories with the same name. History ID's must be unique. Consider this a critical database error.
- **Warning: unknown history entry –** There is a missing history block in the history indicated. Check the history in the GUI and make sure that the history exists.
- **ERROR: CropMan can't find <file name> –** The crops.dat file doesn't exist. Rebuild the crops file from the "Generate Files" page of the GUI and rerun the blockcheck routine.
- **Warning: unknown crop entry -** There is a missing crop block in the history indicated. Check the crops, rotations, and history in the GUI for the block specified and make sure that the correct crop is specified.

Troubleshooting Errors and Warnings

Following is an alphabetical list of the most common errors reported by the modeling system. All other errors should be referred to the corresponding authors for troubleshooting and debugging:

- **ERROR: CropHistory can't find <file name>–** Warning: The system was unable to open the history flat file, histories.dat. Probably this means that the requested file is not present but as with all open failures it can indicate that the file permissions do not allow access or a system failure.
- **ERROR: CropMan can't find <file name>; " -** Warning: The system was unable to open the crop flat file, crops.dat. Probably this means that the requested file is not present but as with all open failures it can indicate that the file permissions do not allow access or a system failure.
- **ERROR: Input line to long -** Most likely one of the .100 files in /usr/local/nrel/LATEST.100 has a line in it that is longer than 80 characters. Edit the file to shorten the line length.
- **ERROR: can't create Events directory -** It is likely that there is a permissions problem in the LINUX computer. If the user has logged in as user "gefsoc" or some other user, and for whatever reason the user has been restricted from adding new directories to /home/gefsoc/model_runs/gefsoc1, then this error may occur. To resolve this error, log in as user "root" and run the command "chmod 777 -R /home/gefsoc", followed by the command "chown -R gefsoc /home/gefsoc".
- **ERROR: could not open rotation file <file name>;** Fatal: The system was unable to open the crop flat file, crops.dat. Then rebuild the crops.dat and histories.dat files and restart the calculation client with the gefeq.pl or gefrun.pl (whichever was running at the time of the error) without issuing the command "srvrcmd abort" or "srvrcmd shutdown".
- **ERROR: fatal file read error –** Fatal Century read error. Check files are not truncated or corrupted.
- **ERROR: finding values in harv.100 -** The user has likely specified a HARV event that is not in the file the active harv.100. Check to make sure the the correct hav.100 id being used and that the HARV option is specified in harv.100., Then restart the calculation.

- **ERROR: history <ID> does not exist -** The user may have specified an incorrect Crop History in the management sequence for this history. Check the Management Sequences to ensure that all of the histories specified exist and that all appropriate chains are specified. Then rebuild the management sequence file and restart the calculation.
- **ERROR: illegal month -** Check each of the crops used in this history to make sure that the months specified for the management activities are all larger or equal to 1 and less than or equal to 12. Then rebuild the "crops.dat" files and restart the calculation client with the gefeq.pl or gefrun.pl (whichever was running at the time of the error) without issuing the command "srvrcmd abort" or "srvrcmd shutdown".
- ERROR: incomplete RothC land data for <year> - There are not 12 months of RothC input data for <year>. Make sure that century is giving monthly output.
- ERROR: IEEE floating exception (NaN or INF) - Century has detected a floating point exception in the internal state. This is a result of bad or extreme inputs causing the simulation to drift. If thisis an equilibrium run or the error refers to the sire file, the effected site file or archive must be deleted. Debugging should be done by a careful he century input and hand executions of the schedule files in the history chain.
- **ERROR: missing CENTURY executable in <directory> -** Check in the directory, should be /usr/local/nrel/models, and ensure that the file Centpl exists there. Also check the file /home/gefsoc/model_runs/gefsoc1/RunFiles.pm to ensure that the correct century model (/usr/local/nrel/models/Centpl) is specified. Note that LINUX is case sensitive for file names.
- **ERROR: missing ID <ID> in history database -** The user specified an undefined Crop History di, <ID>, in the management sequence for this history. Check the Management Sequences to ensure that all specified histories exist. Then rebuild the management sequence file and restart the calculation.
- **ERROR: missing RothC file <file> -** Check in /usr/local/nrel/models and ensure that the specified file exists. Note that LINUX is case sensitive for file names.
- ERROR: missing RothC output file – The RothC output file, graph.dat could not be opened. This indicates that RothC failed.
- ERROR: No results from the RothC equilibrium run - The RothC output file, graph.dat did not contain the equilibrium results. This indicates that RothC failed.
- **ERROR: missing soil texture file <file name> -** The system was unable to open the soil definition file **<file name>** (gefsoils.dat). Probably this means that the requested file is not present but as with all open failures it can indicate that the file permissions do not allow access or a system failure. This error
- means that the weather file **<file name> (**gefsoils.dat) is misspelled or does not exist in the directory /home/gefsoc/model_runs/gefsoc1/. Try regenerating the soils data file and then restart the calculation client with the gefeq.pl or gefrun.pl (whichever was running at the time of the error) without issuing the command "srvrcmd abort" or "srvrcmd shutdown".
- **ERROR: missing climate database <file name> -** The system was unable to open the weather file **<file name>** (gefweather.wth). Probably this means that the requested file is not present but as with all open failures it can indicate that the file permissions do not allow access or a system failure.
- ERROR: no Century C data recorded for RothC – The RothC model was called without the monthly C input data. This is an error in the output specifications for Century.

- **ERROR: unknown crop file entry _ –** There is an error in the structure of the crop flat file, crops.dat. Recreate the file and retry. If contact the corresponding author if the error recurs.
- **ERROR: unknown crop file variable: <variable> -** Variables in the crop.100 file are not in one of the recognized formats. This error indicates that <variable> was found instead of one of the optional variables or the variable following the optional entry. Check the format of crop.100.
- **ERROR: untested OS -** The user is attempting to run the modeling system on an untested OS. The system has been tested onLINUX on the x86, SunOS on sparc, and OS X (Darwin) on PowerPC.
- **ERROR: year less than zero -** Check this history to make sure that the start year and end year are correct for each block. Then rebuild the "histories.dat" file and restart the calculation client.
- **Error reading CO2 flag -** Century error parsing the event file CO2 flag. The input read follows the error message. The CO2 flag specified on the Rotations form for this rotation may be incorrect.
- **Error reading Initial system -** Century error parsing the event file initial system flag. The input read follows the error message. The Initial System specified on the Rotations form for this rotation may be incorrect.
- **Error reading label type -** Century error parsing the event file initial system flag. The input read follows the error message. The label type specified on the Rotations form for this rotation may be incorrect..
- **Error reading output interval-** Century error parsing the event file initial system flag. The input read follows the error message. Check this history to make sure that the block specified has the correct output interval for each of the block- either "monthly" for base, experiment, and the final 500 yr block of the equilibrium histories. The first block of the equilibrium history may be set for "10,000 yr".. Then rebuild the "histories.dat" and "crops.dat" files and restart the.
- **Error: root senescence, rtsen, out of range** Correct the root senescence variable (rtsen) for the current crop so that 0 <= rtsen <.1. This variable is specified in the crop.100 file .
- ERROR: RothC output field overflow – RothC output overflowed the field formats. This probably indicates a failure of the RothC model.
- ERROR: RothC graph.dat output field overflow – RothC output overflowed the field formats. This probably indicates a failure of the RothC model.
- **Fatal Error: Missing site archive -** Century error opening the requested binary site archive The file should have been checked before the Century call so this error indicates corruption of the working directory or am internal error. Delete the temporary directory and restart the run. Contact the corresponding author if it recurs. There is a processing error in the model runs. Rebuild the model input files and restart the calculation client with the gefeq.pl or gefrun.pl (whichever was running at the time of the error) without issuing the command "srvrcmd abort" or "srvrcmd shutdown".
- **Fatal Error: The new binary file exists -** Century error indicating the output binary exists.There is a internal error in the model runs. Delete the temporary directory and restart the run. Contact the corresponding author if it recurs.
- WARNING: skipping <EVENT> in month <MONTH> with no option – There was an event specified without an option from the ".100" file. The user should check all of the

crops in the indicated history or management sequence, rebuild the crops file and then restart the simulation.

- **WARNING: wilting point = field capacity, -** The Century site file for the particular soil indicates that the wilting point and field capacity are the same. This can be a result of not specifying these values and forgetting to specify one of the estimating equations. This may also indicate an internal or system error in the equilibrium runs Check the soil definitions file. and rerun the equilibrium runs. Contacr the corresponding author if the problem recurs.
- **Warning: Fallow called with no events or last month"; -** There appears to be a "FAL" or "FALLOW" crop specified but without any management activities specified. This may be a data entry error. The user should check all of the fallow crops in the indicated history or management sequence, and then rebuild the crops file. Then restart the simulation.
- **Warning: Too many output variables."// & - You** have added too many output variables to the list in /home/gefsoc/model_runs/gefsoc1/RunFiles.pm. Decrease the number and restart.

Post-Processing

(This section is in preparation.)

Appendix 1: Redhat® LINUX Fedora Installation Tutorial

This tutorial was put together to provide step-by-step instructions for installing the GEFSOC LINUX computational server.

The tutorial was assembled by photographing screen shots from an actual LINUX Fedora installation. The quality of the photographs varies but the images should be clear enough to read and understand the steps necessary to complete the installation.

The user may find that there are some differences due to the hardware configuration of the system they are installing. If they do run into differences, they are encouraged to consult their documentation and users manuals and proceed cautiously.

Users may download image (.iso) files for the four CD's necessary for this installation from http://www.nrel.colostate.edu/projects/agecosys/. After users create the installation CDs, they will need to boot the LINUX server from CD #1. If the computer is not currently configured to boot from CD, then the user must modify the BIOS settings for the computer so that it will boot from CD.

Please consult your CD-ROM burning software for instructions on how to produce a CD from a .iso file.

After it is configured to boot from CD, simply put CD #1 into the CD drive, and then reboot and follow the instructions in the images that follow.

Appendix 2: Tutorial on using the Graphical User Interface

Build Crops

1. Click on the "Add New Crop" button at the bottom of the page.

2. Enter a Crop name and description. Ignore the Crop ID number- that is an automatic key field that the database generates for you automatically. Note: For the time being, the Database will allow you to enter crops that aren't in your crop.100 file. However, after the modeling system is installed on your LINUX computers, if you do not have the correct crop in your crop.100 while building your crop histories, you will have to modify your crop.100 file in the following location on the LINUX box: /usr/local/nrel/LATEST.100/crop.100

3. Under "Notes", suggest that you put in your initials and the date, plus any pertinent information that you would like to record that is associated with this crop and need to remember later. By maintaining good notes, you may be able to avoid "what was I thinking????!!!" moments in the future.

4. Begin to enter the crop events. You will need to have events like PLTM, FRST, LAST, SENM, CULT, etc. Basically, you'll need to include anything that you wish Century to do for you. Like note 1.2 above, for the time being you can enter actual events that are not in your .100 files (e.g. harv.100, graz.100) but after the modeling system is installed, you will be limited to working with those events that are in your .100 files.

5. Maintain good notes for all of your events. Insert your initials and the date each is entered.

6. Note that for the cultivation events, most events in the example database are represented by both a "till" event and a "cult" event. We have included a new feature in Century that allows you to parameterize a tillage event specifically to the type of tillage implement that is used. The feature is in a till.100 file (details to follow under separate cover). When this example dataset was created, the till.100 file was used to build the histories but then the till events were converted to equivalent cult events because of bugs in the programming. We have not completely debugged the feature yet, so you should plan to enter only CULT events in your database. The TILL events have been left in for demonstration purposes.

7. Anytime you make a change or enter new information, you may enter the "Save Record" button to ensure the change is immediately saved to the database. It is not generally necessary but it is good insurance to do this.

8. Use the "Delete Activity" and "Duplicate Activity" buttons to create replicates of events/activities within the crop that you have on the screen. Likewise for the crops. In order to efficiently build a set of crops that are similar, we will usually build one crop that is generic and representative, and then duplicate it multiple times and modify the duplicates to match the variations necessary. For example, if you needed to create three rice crops- one for rice-wheat rotations, one for rice-jute-wheat rotations, and one for rice-rice rotations,

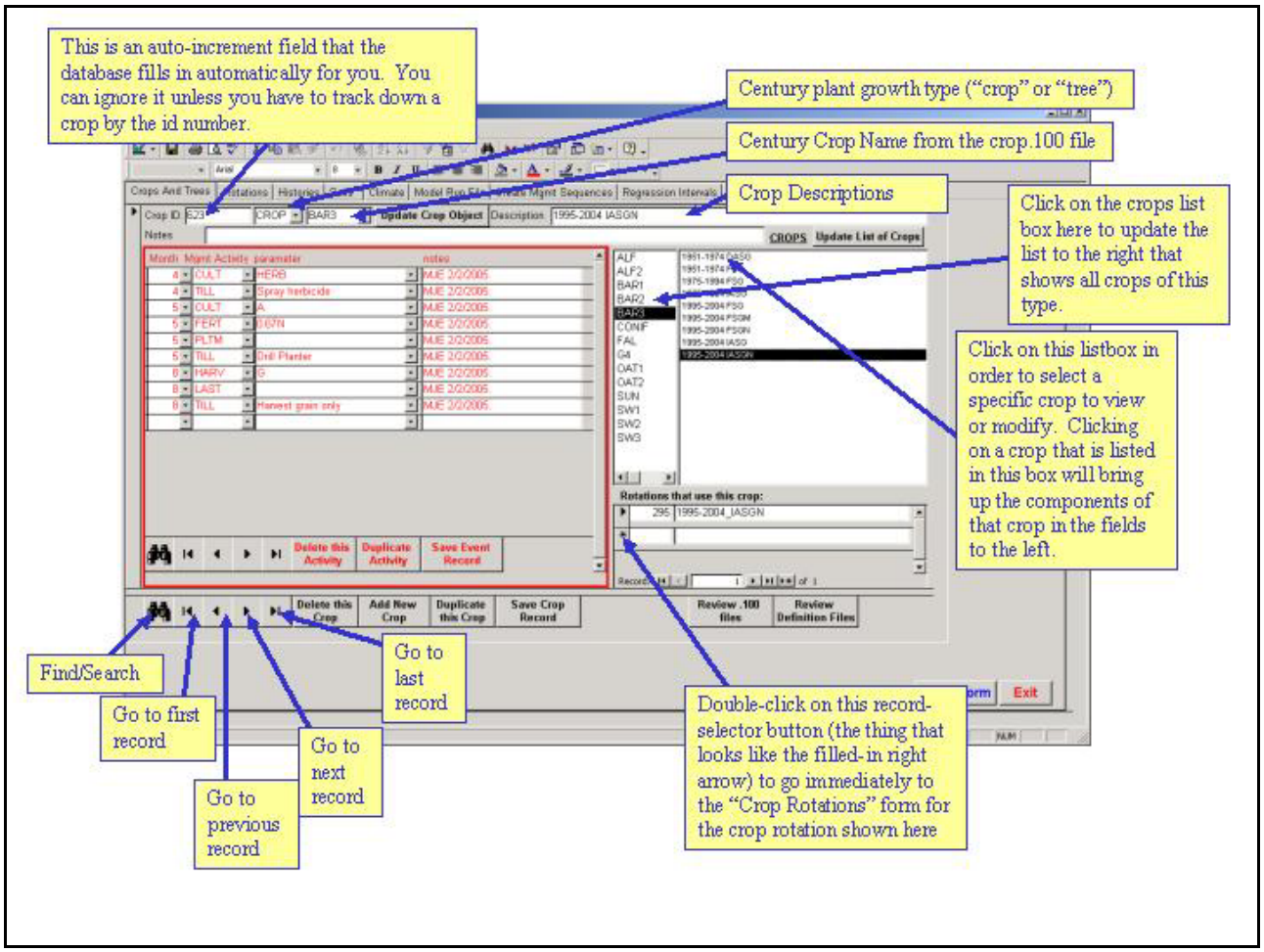
with minor variations in planting/harvesting dates and fertilization values, you could create one rice crop, then duplicate it two times and alter the crop description, fertilizer, and planting/harvesting dates for the two duplicates to match the conditions necessary to simulate the other two crops.

9. After you are finished creating a new crop or altering an existing one, you should do two things immediately:

10. Click on the button "Update List of Crops" in order to make sure the dropdown selection boxes in the rotation field are updated to reflect the new addition.

11. Click on the black "Save Crop Record" button to make sure the database transaction is completed.

12. Finish creating the crops you need for your rotations and crop histories.


This is an auto-increment field that the database fills in automatically for you. You can ignore it unless you have to track down a crop by the id number.
Century plant growth type ("crop" or "tree")
Century Crop Name from the crop.100 file
Crop Descriptions
Click on the crops list box here to update the list to the right that shows all crops of this type.
Click on this listbox in order to select a specific crop to view or modify. Clicking on a crop that is listed in this box will bring up the components of that crop in the fields to the left.
Find/Search
Go to first record
Go to previous record
Go to next record
Go to last record
Double-click on this record-selector button (the thing that looks like the filled-in right arrow) to go immediately to the "Crop Rotations" form for the crop rotation shown here

Build Rotations

1. Now, create a new rotation. Click on the tab/page titled "Crop Rotations" to go to that form. Click on the button "Add New Rotation" at the bottom of the Rotations form.

2. Enter a Rotation name and description. Ignore the Rotation ID number- that is an automatic key field that the database generates for you automatically. Note: Under "Notes", suggest that you put in your initials and the date, plus any pertinent information that you would like to record that is associated with this Rotation and need to remember later. By maintaining good notes, you may be able to avoid "what was I thinking????!!!" moments in the future.

3. Begin to enter the Rotation data. Maintain good notes for all of your crops, where necessary.

4. Under "Rotation Name", suggest you use any short, descriptive name that will allow you to easily differentiate from other histories. For example, a high-fertilizer, conventionally-tilled rice-wheat rotation for the lower IGP might be named "rice-wheat-lowerIGP-highN-CT".

5. Under rotation description, provide a more detailed description of the rotation that is not provided in the rotation name.

6. Anytime you make a change or enter new information, you may enter the "Save Record" button to ensure the change is immediately saved to the database. It is not generally necessary but it is good insurance to do this.

7. To add your first crop to the rotation, click on the purple "Add Crop to this Rotation" button on the right side of the form. This will create a blank space for you to add a crop to the rotation. Then, click on the purple dropdown box labeled "crop" in the top center of the form, and a list of all of the crops you have created in the crop form should appear. NOTE: If those crops don't appear for whatever reason, go back to the "Crops" form and click on the "Update List of Crops" button, or alternatively, click in the drop down box once, then click on the "Records" menu and select "Refresh".

8. To add a new crop that comes after this crop, click on the "Add Crop to this Rotation" button again. IF it is the same crop that is grown two years in a row, you can click on the button "Duplicate Crop in this Rotation". To delete a crop from the rotation, simply click on the "Remove Crop from this Rotation" button. It will be pulled from the sequence of crops, but the crop will not be deleted from the Crops table.

9. NOTE: The "order" field orders the crops within the rotation. It is maintained in a relatively superficial manner by the database. IT is NOT necessary for the numbers in the crop order to be exactly sequential- you simply want them to sort correctly. So if you have two crops, one with a order number of 3 and the next with 5, as long as you intend for crop 5 to come after crop 3, then there is no reason to update it. Here are some more details on how the order field is maintained by the database.

10. If you add a new crop, the crop is appended to the end of the list of crops, and the order field is set to the highest existing number in the rotation plus one.

11. IF you remove a crop from the rotation, the crop is removed and all of the order fields for the crops later in the sequence than the one removed are decremented by 1.

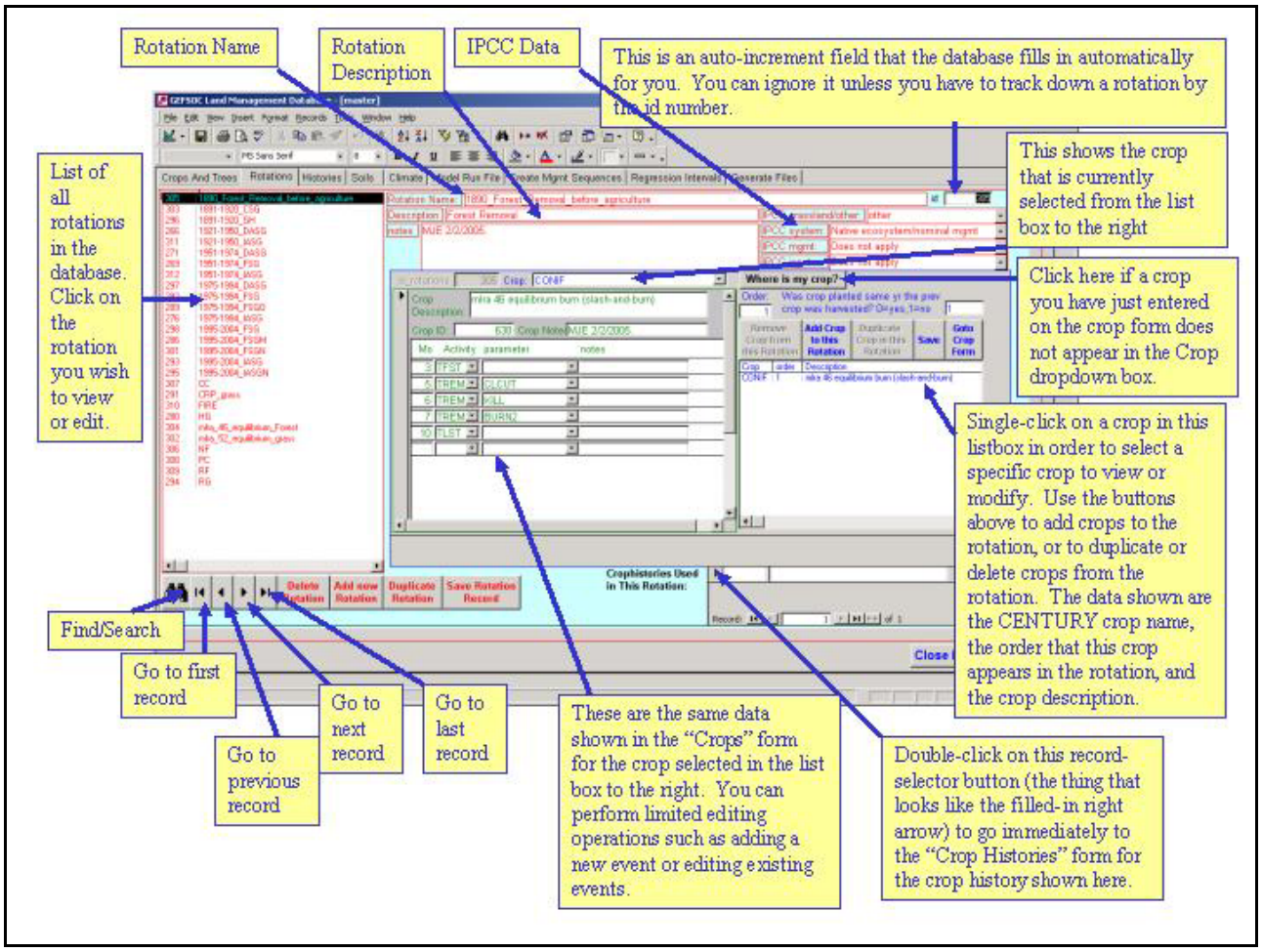
12. IF you duplicate a crop from the rotation, the crop is added immediately after the crop you are duplicating, and all of the order fields for the crops later in the sequence are incremented by 1).

13. You may also modify the crop order manually by changing the crop order in the "order" field.

14. IMPORTANT NOTE ABOUT CROPS THAT ARE GROWN IN MORE THAN ONE CALENDAR YEAR. Century is designed to manage crop growth by the calendar year. There are a number of good programming reasons to do this but unfortunately it creates some problems when you try to generate schedule files. SO, it is critical to indicate if you are double- or triple-cropping within a single year, or planting winter wheat in say November and harvesting it in May. You do this by entering a "0" (zero) in the field titled "Was crop planted same yr the prev. crop was harvested?". For example, if you have a rice-jute-wheat rotation planted as follows: rice planted in March, harvested June, Jute planted July, Harvested September, Winter Wheat planted October, harvested in February, you would enter a "1" for the rice crop, and a "0" for the Jute and Winter Wheat crops.

15. Similarly, if you have one crop that is very long growing, such as a fallow-wheat crop that involves two calendar years- you would enter a "1" in this field, but you can enter events that run into following calendar years. An example would be a planting operation in May of the the year following the fallow period would be entered as month 17.

16. After you are done Building rotations, go to assembling the crop histories.


Rotation Name
Rotation Description
IPCC Data
This is an auto-increment field that the database fills in automatically for you. You can ignore it unless you have to track down a rotation by the id number.
This shows the crop that is currently selected from the list box to the right
List of all rotations in the database. Click on the rotation you wish to view or edit.
Click here if a crop you have just entered on the crop form does not appear in the Crop dropdown box.
Single-click on a crop in this listbox in order to select a specific crop to view or modify. Use the buttons above to add crops to the rotation, or to duplicate or delete crops from the rotation. The data shown are the CENTURY crop name, the order that this crop appears in the rotation, and the crop description.
Find/Search
Go to first record
Go to previous record
Go to next record
Go to last record
These are the same data shown in the "Crops" form for the crop selected in the list box to the right. You can perform limited editing operations such as adding a new event or editing existing events.
Double-click on this record-selector button (the thing that looks like the filled-in right arrow) to go immediately to the "Crop Histories" form for the crop history shown here.

Build Histories

1. Now, create a new crop history. Click on the tab/page titled "Crop Histories" to go to that form. Click on the button "Add New crop history" at the bottom of the crop history form.

2. Enter a crop history name and a description. Ignore the crop history ID number- that is an automatic key field that the database generates for you automatically. Note: Under "Notes", suggest that you put in your initials and the date, plus any pertinent information that you would like to record that is associated with this Rotation and need to remember later. NOTE that the base and equilibrium history names must correspond to those defined in the model run table.

3. Maintain good notes for all of your histories, where necessary. By maintaining good notes, you may be able to avoid "what was I thinking????!!!" moments in the future.

4. Under "History Name", suggest you use any short, descriptive name that will allow you to easily differentiate from other histories. For example, a high-fertilizer, conventionally-tilled rice-wheat rotation for the lower IGP might be named "rice-wheat-lowerIGP-highN-

CT". We typically use the same or similar crop history names as we do for the Rotations that are associated with them, with modifications associated with their time period in the history or subtle variations in management.

5. Under crop history description, provide a more detailed description of the history that is not provided in the crop history name.

6. Anytime you make a change or enter new information, you may enter the "Save Record" button to ensure the change is immediately saved to the database. It is good insurance to do this.

7. To add your first rotation to the crop history, note that in the set of boxes labeled in red on the right side of the form, there is a blank set of records. Click on the dropdown box for the Rotation field, and you will find a complete list of all of the rotations in the database. If you cannot find in the list a rotation that you know you've already created, click once in the rotation field on this page so that the cursor is in the field. Then click on the "Records" menu and then the "Refresh" option. Then try clicking on the dropdown box again. Select the rotation you wish to add to the crop history.

8. Then, update the start year and end years in the history block. NOTE: The start year of one block should be one year later than the end year of the block that precedes it!

9. Then, indicate whether the rotation in this block is irrigated (I) or dryland (D), and select the IPCC tillage type. The "Description" field is provided for reference purposes and cannot be edited from this form.

10. Finally, enter the output interval for the block. In order to provide the data necessary to run the RothC model, the output interval for all blocks associated with base histories and experiment runs must be monthly, or 0.0833 (which is one divided by 12). The output interval for equilibrium runs is described later in this section.

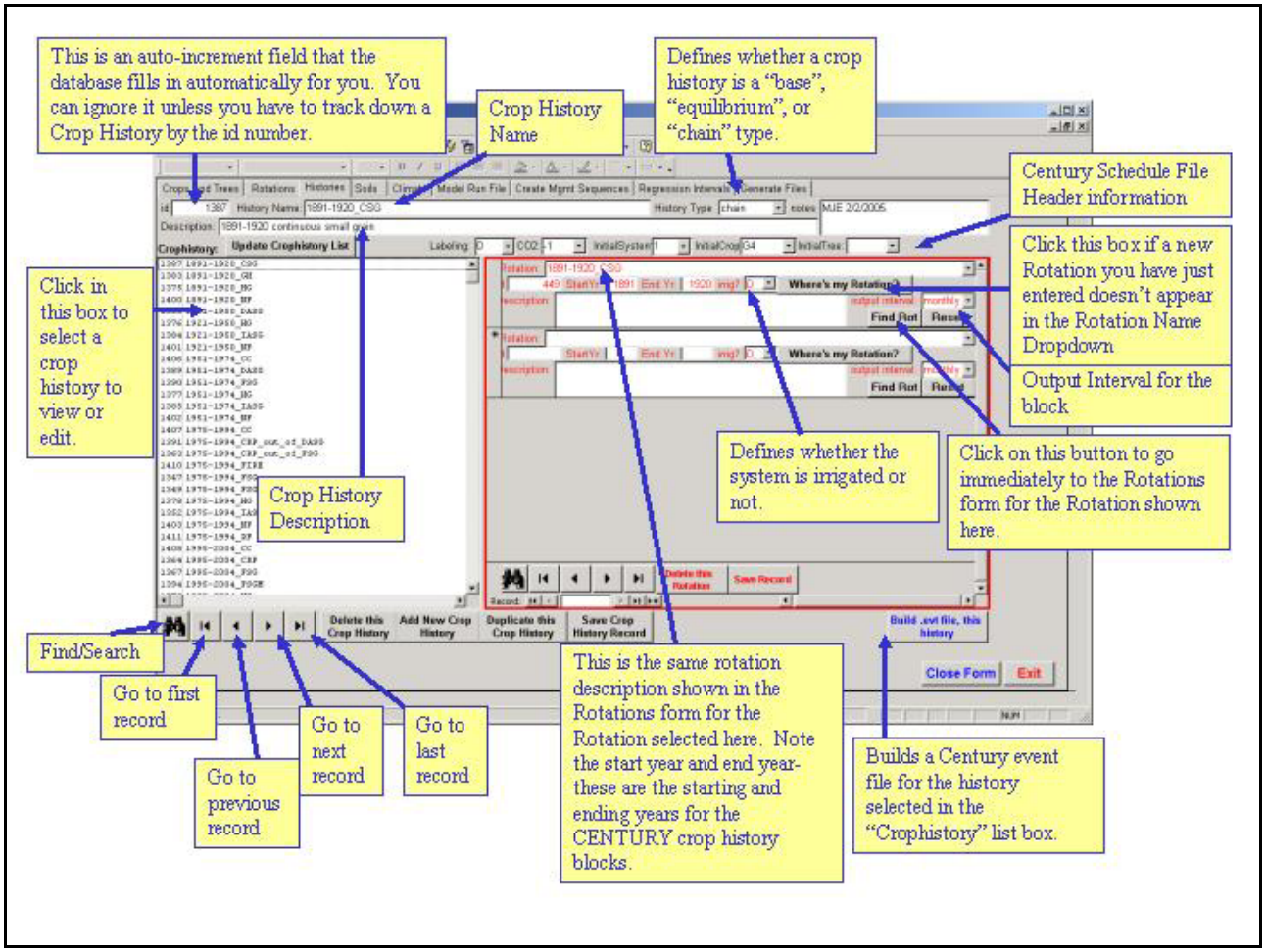
11. Please NOTE that the rotations that are listed for each crop history are sorted according to their start year.

12. NOTE REGARDING CREATING EQUILIBRIUM RUNS. There are several key requirements that your equilibrium runs will have to meet:

13. The name of the equilibrium runs must correspond to the names defined in the run table under the column "Equilibrium".

14. There must be at least two blocks - the first must be for a time period appropriate for the equilibrium history (e.g. 7,000-10,000 years, or the time period over which the natural vegetation has been relatively stable and during which the soil carbon reaches a relatively stable condition). The output interval for this period must be set to 10,000 years, essentially to produce the output at the end of the period. This is to shorten the run time and provide the end point data required to initialize the second block.

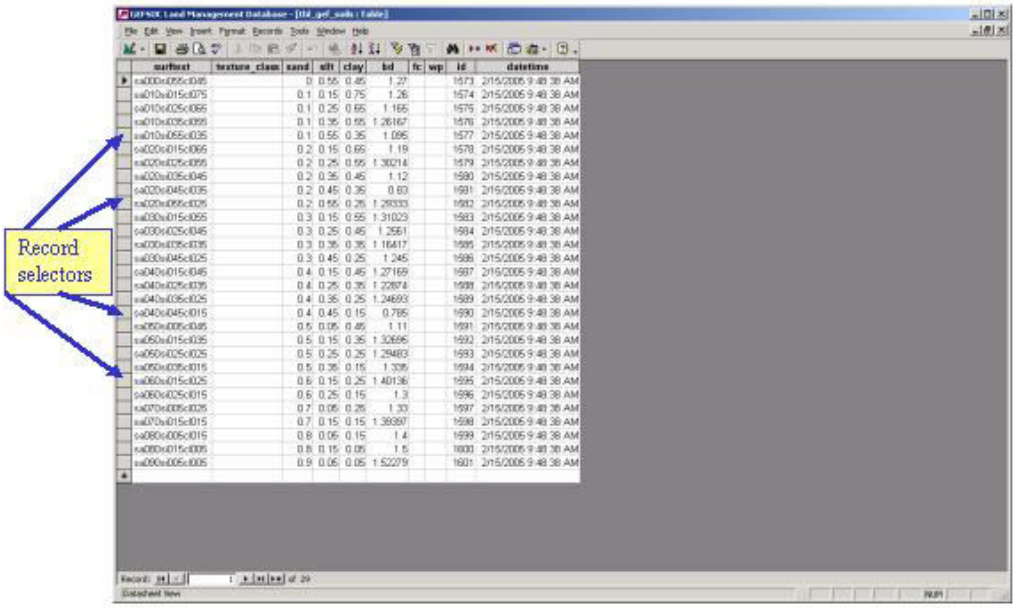
15. There must be a block at the end of the equilibrium period that runs for 500 years, and which is set to produce output at a monthly output interval for that period. These data are used to calculate the average monthly Ccinputs for the RothC equilibrium run.


This is an auto-increment field that the database fills in automatically for you. You can ignore it unless you have to track down a Crop History by the id number.
Crop History Name
Defines whether a crop history is a "base", "equilibrium", or "chain" type.
Century Schedule File Header information
Click this box if a new Rotation you have just entered doesn't appear in the Rotation Name Dropdown
Output Interval for the block
Click in this box to select a crop history to view or edit.
Crop History Description
Defines whether the system is irrigated or not.
Click on this button to go immediately to the Rotations form for the Rotation shown here.
Find/Search
Go to first record
Go to previous record
Go to next record
Go to last record
This is the same rotation description shown in the Rotations form for the Rotation selected here. Note the start year and end year- these are the starting and ending years for the CENTURY crop history blocks.
Builds a Century event file for the history selected in the "Crophistory" list box.

Build Soils Classification Dataset

The modeling system requires that all soils in the SOTER database be classified using a linear classification scheme, as described on page 17 in the Soils section. The soils dataset are generated by defining the desired number of classification divisions in the "Division of Soil Texture Classes (Suggest 10)" field of the "Generate Tables" page, specifying where the SOTER database resides, and then clicking on the "Soils" button in the lower left corner of the "Generate Tables" page. This will generate the lookup table necessary for the modeling system.

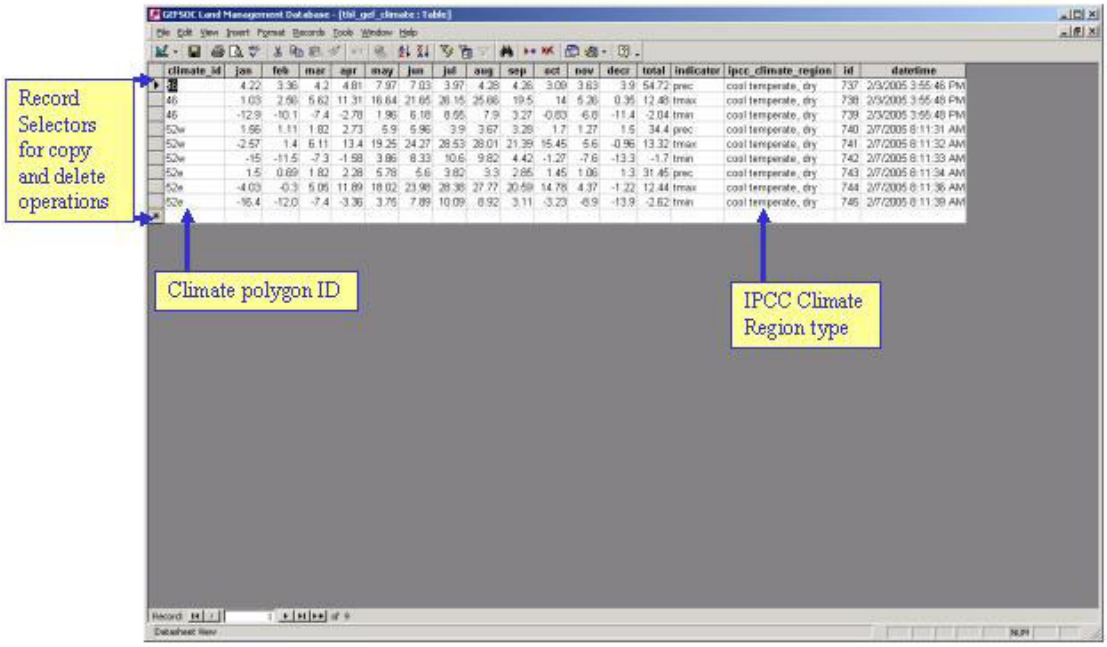
Once this is done, no editing of the soils datasets is necessary.



Climate Data

Users must combine the climate data necessary for Century with IPCC climate classifications for each of the climate regions in the database. Those classifications are described in the table named "tbl_ipcc_climate_regions" in the database. To view the tables, click on the "Windows" menu and then select the option labeled "gefsoc_landmgmt_database_woutODBC". Then click on the "tables" tab on the left side of the screen and double-click on the icon for the table named tbl_ipcc_climate_regions.

Users may paste your climate data into the form from Excel or text files as long as columns are tab-delimited and the column order in the Excel or text file matches exactly with the column order in the database table.



Model Run Table

The Model Run Table is the crux of the modeling system. It is the file that links all of the users data together for the LINUX system to actually run the models.

We recommend using a GIS to build the run table into a tab-delimited format. The run table is essentially the unique combinations of the intersections of your GIS layers for climate, soil classification, natural vegetation (equilibrium run type), latitude of climate polygon center, longitude of climate polygon center, and land management sequence.

land_management_unit: The land management unit defined from the base land use polygons.

climate: The climate region identifier for your climate classification.

id_mgmt Sequence: This defines the unique sequence over time of the management changes that occur on the site, running from the base period through the future scenarios. We have found, at NREL, that the easiest way at NREL for us to label these is to simply string together a series of

management codes from the earliest period to the latest period, separated by hyphens or underscores. For example, the code "FW-FW-FW-FW" would indicate the following:

- Base period: fallow-wheat
- Recent period: fallow-wheat
- Current period: fallow-wheat
- Future period: fallow-wheat

Likewise, the code "FW-FW-IC-IC" would indicate the following:

- Base period: fallow-wheat
- Recent period: fallow-wheat
- Current period: irrigated corn
- Future period: irrigated corn

id_equil: This defines the equilibrium history as defined by the potential natural vegetation for the site. The entry in this column must be exactly the same name as the equilibrium history name in the database to which this run corresponds.

drain1: If the region you are modeling was drained early in its history using non-mechanized (e.g. ditch) drainage techniques, then list the year the represents the midpoint of the time period over which the drainage occurred. This should only be done for hydric soils.

drain2: If the region you are modeling was drained early in its history using mechanized (e.g. tile) drainage techniques, then list the year the represents the midpoint of the time period over which the drainage occurred. This should only be done for hydric soils.

soterunit: Surface texture classification corresponding to the texture types classified on the soils page.

area: Land area associated with the intersection of the base land use, climate, potential natural vegetation, and soils polygons.

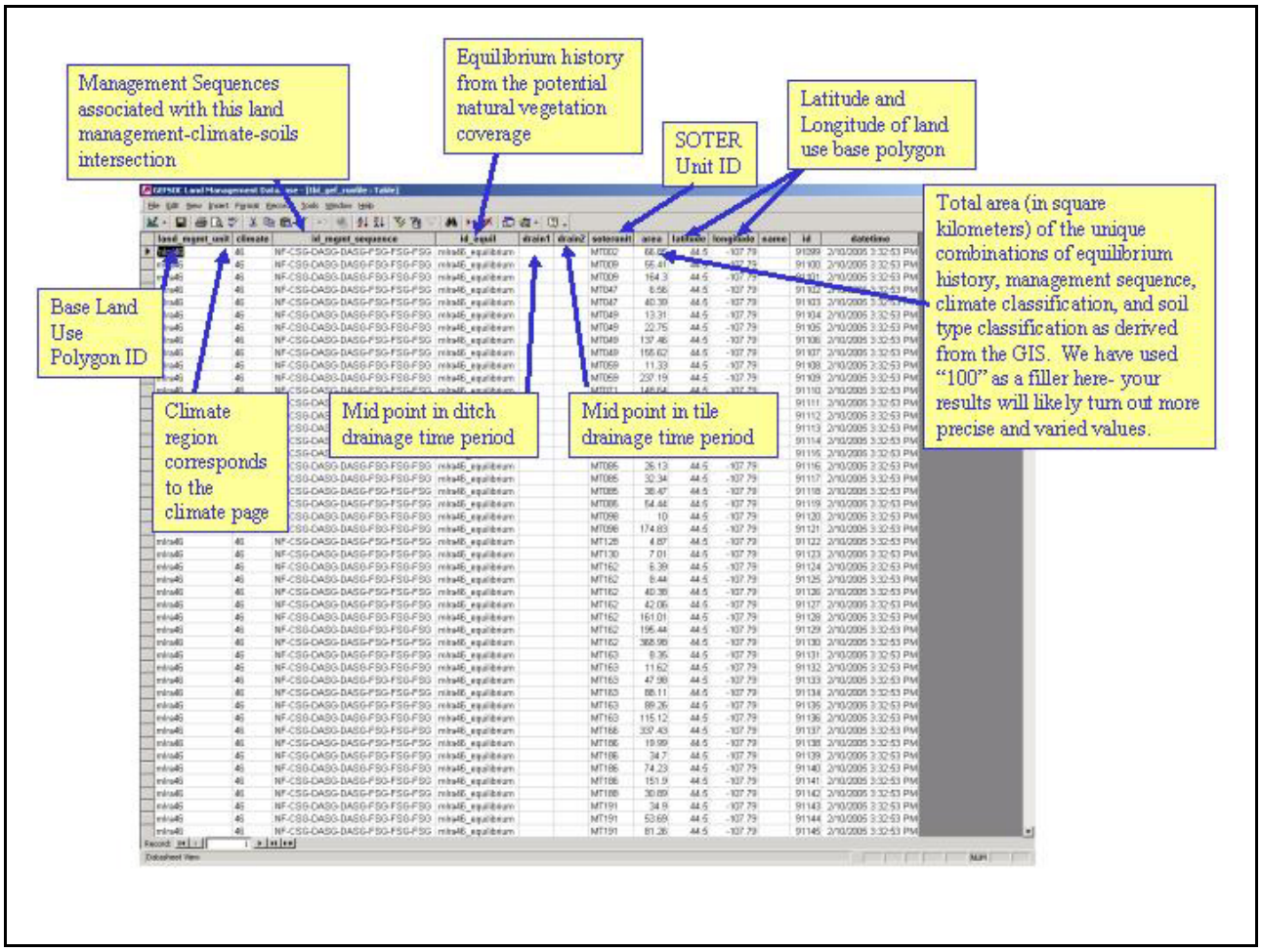
latitude: Latitude of the centroid of the base land use polygon.

longitude: Longitude of the centroid of the base land use polygon.

name: A "user friendly" name that may be attached to this individual record. It can follow any format up to 50 characters long.

id: This field is for system purposes and is not user-editable.

datetime: Timestamp for this record- not user-editable.


Management Sequences associated with this land management-climate-soils intersection
Equilibrium history from the potential natural vegetation coverage
SOTER Unit ID
Latitude and Longitude of land use base polygon
Total area (in square kilometers) of the unique combinations of equilibrium history, management sequence, climate classification, and soil type classification as derived from the GIS. We have used "100" as a filler here- your results will likely turn out more precise and varied values.
Base Land Use Polygon ID
Climate region corresponds to the climate page
Mid point in ditch drainage time period
Mid point in tile drainage time period

Management Sequences

The management sequences page allows users to link specific cropping histories with management sequences identified in the model run table. The cropping histories for each individual chain in the management sequence are linked by sequentially linking the histories that the user has entered into the database.

Explanations for each data field are as follows:

Land Management Unit: This provides a list of all of the unique land management units in the Model Run Table, listed alphabetically. The user selects one of these to bring up a list of all of the unique management sequences for that land management unit.

Management Sequence: Select one of these management sequences in order to view/edit the area weighting factor or base/chain crop histories associated with it.

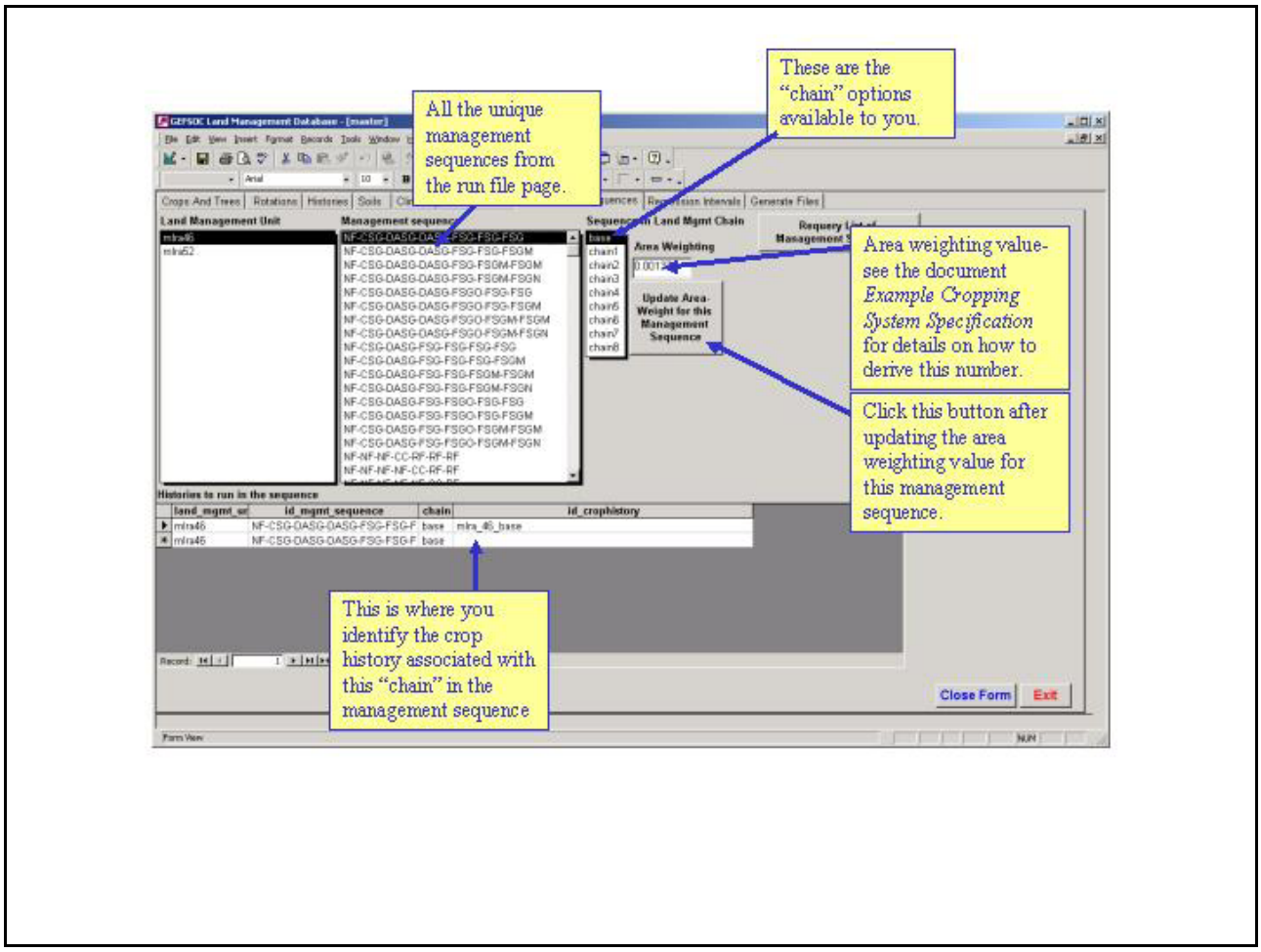
Land Management Unit: This provides a list of all of the unique land management units in the Model Run Table, listed alphabetically. The user selects one of these to bring up a list of all of the unique management sequences for that land management unit .

Sequence in Land Mgmt Chain: There is one base and up to eight chain histories possible that may be sequentially linked in the management sequence chain. The base and chains are linked consecutively by the modeling system. In the Example analysis provided with this document, for the land management unit of mlra 46 and management sequence id of NF-CSG-DASG-DASG-FSG-FSG-FSG, there is one base history and six chains, as follows:

Base/Chain	**Time Period**
Base: NF (native forest)	1880-1890
Chain1: CSG (continuous small grains)	1890-1920
Chain2: DASG (dryland alfalfa-small grains)	1921-1950
Chain3: DASG (dryland alfalfa-small grains)	1951-1975
Chain4: FSG (fallow-small grains)	1976-1995
Chain5: FSG (fallow-small grains)	1996-2004
Chain6: FSG (fallow-small grains)	2005-2030

Area Weighting: This is where the user may enter the area-weighting factor associated with this specific land management unit- management sequence combination, from the management sequence diagrams (Figure 13 and Figure 14).

Histories to Run in the Sequence: The user may identify one history for each base or chain in the management sequence. The field provides a dropdown/list box of the histories entered in the Histories page of the GUI.

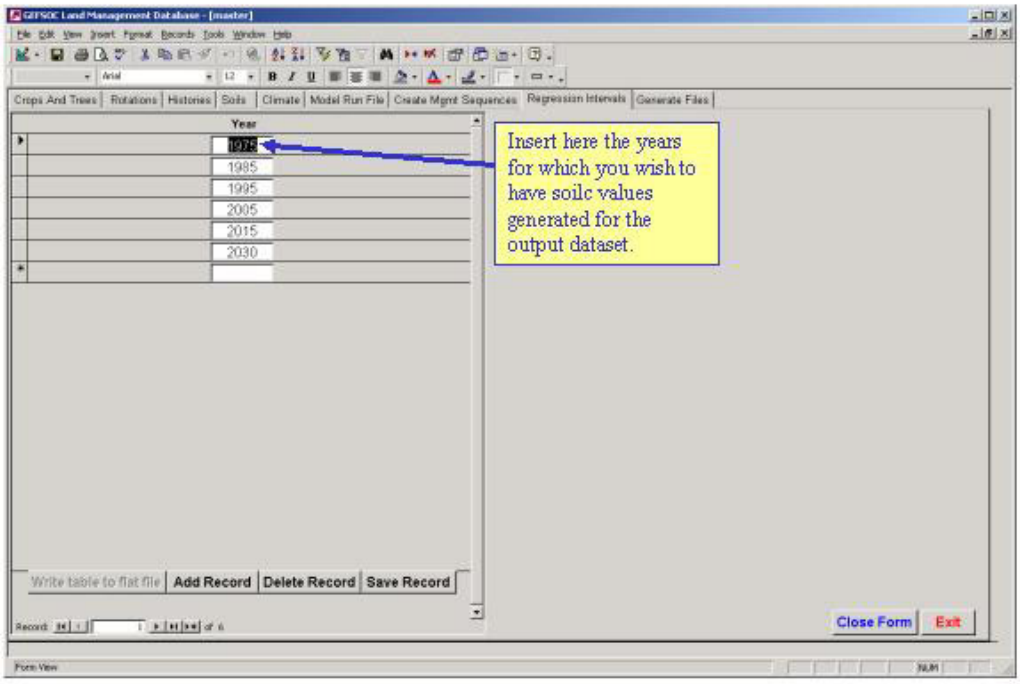

All the unique management sequences from the run file page.
These are the "chain" options available to you.
Area weighting value- see the document *Example Cropping System Specification* for details on how to derive this number.
Click this button after updating the area weighting value for this management sequence.
This is where you identify the crop history associated with this "chain" in the management sequence
Close Form
Exit

Regression Intervals

The modeling system provides output for specific regression intervals during the modeling period. This is done to reduce the size of the output dataset, as described in the section titled *Regression Intervals* on page 28.

The data entry for this section is straightforward. Simply enter the years for which you would like to know the soil C values output from the model runs. You may enter as many years as you like, however entering too many years defeats the purpose of reducing the size of the output dataset. We recommend outputting data at decade intervals or at points where important baseline statistics are desired or where critical land use changes were experienced over large areas. We do not recommend exporting regression statistics for periods that span history blocks- for example, if history chain 4 in a certain management sequence begins in 1980 and ends in 1989, and history chain 5 begins in 1990 and ends in 2004, we would not recommend entering a regression interval from 1985 to 1995. This is due to the fact that the Century model (rarely, but it still happens) produces spurious carbon input values at the break point years between history blocks.

See the above-mentioned section for details on how the regression statistics are generated.



Generate Files

Before the user can initiate the model runs, he or she must produce the input datasets necessary for the modeling system. This page contains the functions necessary to generate those files based on the data the user has input via the GUI.

There are two sections to the page:

- Modeling System Defaults (fields on the top half of the form)
- Generate Files and Run Models Buttons (fields on the bottom half of the form)

Defaults

Text File Editor: This is the text file editor that will be used to show the contents of the various modeling system input files generated by the GUI. We recommend Textpad© or VIM, although Microsoft® Notepad or Wordpad will work.

Location of model run files and scripts: You must specify the subdirectory on the LINUX machine under the /home/gefsoc/model_runs/ where the modeling system input files are installed. Default is "gefsoc1".

Location of Windows File Manager/Explorer: In order to bring up the Century model .100 values and .def files in the crops form, the user must specify where the Windows Explorer (NOT the Internet Explorer) is located. This is the same as the Windows File Manager.

Drive mount connected to /home/gefsoc/: This specifies the windows drive letter that the user has mounted to the location /home/gefsoc on the LINUX computer.

Drive mount connected to /usr/local/nrel/: This specifies the windows drive letter that the user has mounted to the location /usr/local/nrel on the LINUX computer.

IP address of MySQL Server (probably 10.10.11.101): User must identify the IP address of the LINUX computer connected to the PC.

MySQL username: The default for this field is 'gefsoc', as that is the username contained in the permissions tables shipped with the modeling system.

MySQL database password: The default for this field is 'gefsoc', as that is the password associated with username = 'gefsoc' contained in the permissions tables shipped with the modeling system.

MySQL database name: The default for this field is 'gefsoc', as that is the database name for the database tables shipped with the modeling system. Users may update this to any value they wish as long as they duplicate the MySQL table structures in the new database.

Irri.100 option to use for automatic irrigation (suggest A95): Users may specify that history blocks are automatically irrigated on the Histories page of the GUI. This option tells the modeling system which irrigation option from the Century irri.100 file to use when automatic irrigation is specified.

Path to SOTER database and database name: In order to add in the SOTER soils database, the user must specify the complete path to the SOTER database.

Generate Files and Run Models Buttons

Histories: The cropping histories entered into the "Histories" page must be exported to a format the modeling system can use. The default file name for this file is /home/gefsoc/model_runs/gefsoc1/histories.dat.

Crops: The rotations and crops entered into the "Rotations" and "Crops and Trees" pages must be exported to a format the modeling system can use. The default file name for this file is /home/gefsoc/model_runs/gefsoc1/crops.dat.

Run Checkblock: This runs the checkblock routine describe in the section titled *Running the* Century *Block Check routine*. The output dataset from the checkblock routine is /home/gefsoc/model_runs/gefsoc1/checkblock.txt

Soils: This generates the soils data from the SOTER database. The output dataset is the table tbl_gef_soils, written to the file /home/gefsoc/model_runs/gefsoc1/gefsoils.dat.

Climate: This builds the climate file /home/gefsoc/model_runs/gefsoc1/gefweather.wth from the table tbl_gef_climate.

Model Run File: Builds the model run file /home/gefsoc/model_runs/gefsoc1/gefregion.dat.

Mgmt Sequences: Builds the management sequences file /home/gefsoc/model_runs/gefsoc1/gefexp.txt

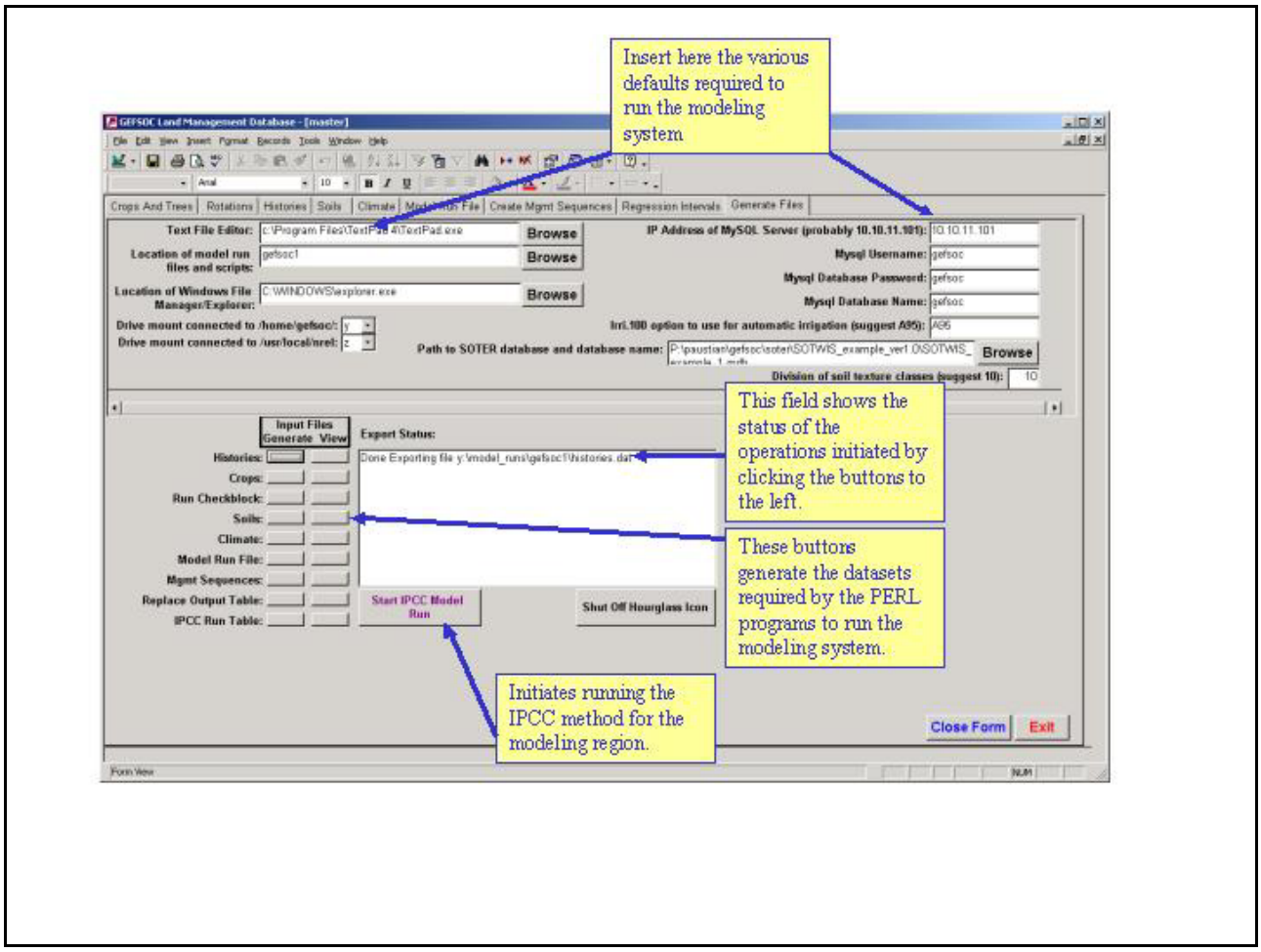
Replace Output Table: The output dataset for the modeling system must be rebuilt after the regression intervals are first entered into the GUI, or if they are modified anytime afterward. The output dataset is tbl_gef_output.

IPCC Run Table: Before running the IPCC method for the modeling region, the user must generate the IPCC run table. The table is tbl_ipcc_runfile.

Start IPCC Model Run: This button will start the process of running the IPCC method for the modeling region. It will warn the user about the estimated time length prior to starting the model run.

Shut Off Hourglass Icon: Occasionally the user will run into errors that occur partway through a process. When that happens, the hourglass icon may remain in place of the mouse pointer after the user halts the process. Click on this button to restore the mouse pointer.

Export Status: Shows the status of processes started from this page.


Insert here the various defaults required to run the modeling system
This field shows the status of the operations initiated by clicking the buttons to the left.
These buttons generate the datasets required by the PERL programs to run the modeling system.
Initiates running the IPCC method for the modeling region.
Browse
Input Files Generate View
Export Status:
Start IPCC Model Run
Shut Off Hourglass Icon
Close Form
Exit

Appendix 3: Modeling System Installation Script for LINUX

```
#########################################################
### USER MUST LOG IN AS ROOT TO EXECUTE THIS SCRIPT ###
#########################################################

######## Set the Shell Spec ########
#! /bin/bash

######## Add Users ########
groupadd mysql
useradd -g mysql mysql
useradd -p '$1$nPDlqpNO$VFsGvtKTYvDWdqzmAwWaX1' -c "GEFSOC user" gefsoc

######## CREATE GEFSOC DIRECTORY STRUCTURE ########
rm -Rf /usr/local/nrel
mkdir -p /usr/local/nrel
mkdir -p /home/gefsoc/model_runs/gefsoc1

######## COPY OVER FILES ########
#copy over installation files
cp -Ra /mnt/cdrom/nrel/* /usr/local/nrel/.
#copy over home directory files
```

```
cp -Ra /mnt/cdrom/gefsoc/model_runs/gefsoc1/* /home/gefsoc/model_runs/gefsoc1/.

######## Configure and start the TELNET service ########
rpm -e /usr/local/nrel/installation_files/telnet-server-0.17-25.i386.rpm
rpm -i /usr/local/nrel/installation_files/telnet-server-0.17-25.i386.rpm

######## Configure and start the RSH service ########
rpm -e /usr/local/nrel/installation_files/rsh-server-0.17-21.i386.rpm
rpm -i /usr/local/nrel/installation_files/rsh-server-0.17-21.i386.rpm
cp -Ra /usr/local/nrel/configuration_files/rsh /etc/pam.d/.

######## Set up security to run the network communications #######
echo rexec >> /etc/securetty
echo rlogin >> /etc/securetty
echo rsh >> /etc/securetty
echo telnet >> /etc/securetty
/sbin/chkconfig --level 345 rsh on
/sbin/chkconfig --level 345 rexec on
/sbin/chkconfig --level 345 rlogin on
/sbin/chkconfig --level 345 telnet on

cp /usr/local/nrel/configuration_files/hosts.allow /etc/.
cp /usr/local/nrel/configuration_files/hosts.deny /etc/.
cp /usr/local/nrel/configuration_files/hosts.equiv /etc/.
cp /usr/local/nrel/configuration_files/hosts /etc/.
cp /usr/local/nrel/configuration_files/.rhosts /home/gefsoc/.
cp /usr/local/nrel/configuration_files/rsh /etc/xinit.d/.

/etc/init.d/xinetd reload
/etc/init.d/xinetd restart

######## Install the MySQL server ########
cd /usr/local/nrel/installation_files
gunzip ./mysql-4.0.21.tar.gz
tar -xvf ./mysql-4.0.21.tar
cd mysql-4.0.21
CFLAGS="-O3" CXX=gcc CXXFLAGS="-O3 -felide-constructors -fno-exceptions -fno-
rtti" ./configure --prefix=/usr/local/mysql --enable-assembler --with-mysqld-
ldflags=-all-static
make
make install
cp --reply=yes support-files/my-medium.cnf /etc/my.cnf
cd /usr/local/mysql
/usr/local/mysql/bin/mysql_install_db --user=mysql
chown -R root /usr/local/mysql
chown -R mysql.mysql /usr/local/mysql/var
chmod -R 777 /usr/local/mysql/var/

cd /usr/local/bin
ln -s /usr/local/mysql/bin/* .
cd /usr/local/lib
ls -s /usr/local/mysql/lib/* .
cd /usr/local/include
ln -s /usr/local/mysql/include/* .

######## Install the MySQL C++ API ########
cd /usr/local/nrel/installation_files
```

```
gunzip ./mysql++-1.7.17.tar.gz
tar -xvf ./mysql++-1.7.17.tar
cd mysql++-1.7.17
CFLAGS="-O3" CXX=gcc CXXFLAGS="-O3 -felide-constructors -fno-exceptions -fno-
rtti" ./configure --prefix=/usr/local/mysql --enable-assembler --with-mysqld-
ldflags=-all-static
make
make install

######## Install the MySQL DBD ########
cd /usr/local/nrel/installation_files
gunzip ./DBD-mysql-2.9004.tar.gz
tar -xvf ./DBD-mysql-2.9004.tar
cd DBD-mysql-2.9004
perl Makefile.PL
make
make install

######## Install the MySQL DBI API ########
cd /usr/local/nrel/installation_files
gunzip ./DBI-1.45.tar.gz
tar -xvf ./DBI-1.45.tar
cd DBI-1.45
perl Makefile.PL
make
make install

######## Install the PERL Time::HiRes.pm module ########
cd /usr/local/nrel/installation_files/Time-HiRes-1.65
gunzip ./Time-HiRes-1.65.tar.gz
tar -xvf ./Time-HiRes-1.65.tar
cd Time-HiRes-1.65
perl Makefile.PL
make
make install

#### update the source library path ####
cd /usr/local/lib/
ln /usr/local/mysql/lib/mysql/* .
echo /usr/local/lib >> /etc/ld.so.conf
/sbin/ldconfig

#### COPY OVER THE MySQL SERVER STARTUP SCRIPT####
cp -a /usr/local/nrel/configuration_files/mysql.server /etc/init.d/.

chmod 744 /etc/init.d/mysql.server
chown -R mysql.mysql /usr/local/mysql/var/
ln -s /etc/init.d/mysql.server /etc/rc3.d/S92mysql.server
ln -s /etc/init.d/mysql.server /etc/rc3.d/K92mysql.server
ln -s /etc/init.d/mysql.server /etc/rc5.d/S92mysql.server
ln -s /etc/init.d/mysql.server /etc/rc5.d/K92mysql.server
/etc/init.d/mysql.server start

#### COPY OVER THE MySQL PERMISSIONS TABLES ####
/etc/init.d/mysql.server stop
# remove existing mysql permissions tables
rm /usr/local/mysql/var/mysql/*
```

```
# replace with permission tables that contain root and gefsoc permissions
cp -a /usr/local/nrel/mysql/mysql_permissions_dir/* /usr/local/mysql/var/mysql/.
# replace with gefsoc database tables
mkdir /usr/local/mysql/var/gefsoc
cp -a /usr/local/nrel/mysql/gefsoc/* /usr/local/mysql/var/gefsoc/.
/etc/init.d/mysql.server start

######## Set the file nrel.gefsoc.modified for future reference ########
touch /etc/nrel.gefsoc.modified

######## Configure and start the SAMBA service ########
cp /usr/local/nrel/configuration_files/smb.conf /etc/samba/.
#Need to add gefsoc user
#User should type the word "gefsoc" when queried for the password
smbpasswd -a gefsoc
/etc/init.d/smb start
cd /etc/rc3.d/
ln -s /etc/init.d/smb /etc/rc3.d/K35smb
cd /etc/rc5.d/
ln -s /etc/init.d/smb /etc/rc5.d/S91smb

######## Configure and start the NFS service ########
cp /usr/local/nrel/configuration_files/exports /etc/.
/etc/init.d/nfs start
cd /etc/rc3.d/
ln -s /etc/init.d/nfs /etc/rc3.d/K20nfs
ln -s /etc/init.d/nfslock /etc/rc3.d/S14nfslock
cd /etc/rc5.d/
ln -s /etc/init.d/nfs /etc/rc5.d/K20nfs
ln -s /etc/init.d/nfslock /etc/rc5.d/S60nfslock

######## UPDATE permissions ########
chmod -R 777 /usr/local/nrel
chmod -R 777 /home/gefsoc
chown -R gefsoc /usr/local/nrel
chown -R gefsoc /home/gefsoc

######## Add a link from /usr/local/mysql/var/gefsoc to /usr/local/nrel ########
ln -s /usr/local/mysql/var/gefsoc /usr/local/nrel/mysql_gefsoc_data

echo
################################################################################
###############
echo ### Check first to see if mysql server is up by running mysql --user=root -
-password=cosfeg ###
echo ### If you can't connect, then
###
echo ### Remember to Start up MySQL server from /etc/init.d/mysql.server start
###
echo
################################################################################
###############
```

Appendix 4: Soil Texture Classification Function

```
Function soilclass(fSand As Single, fSilt As Single, fClay As Single, iNumInt As Integer)
    '*********************************************************************
    '* subroutine:  soilclass
    '* purpose: Classifies soil texture according to a set number of
    '* classification intervals passed as the 4th argument to the function.
    '* author: M. Easter, 22 Dec 2004
    '* Natural Resource Ecology Laboratory, CSU
    '*********************************************************************
    '* Variables:
    '*      iFrac       generic soil fraction variable (integer)
    '*      fSiltClass  silt class (float/single)
    '*      fSandClass  sand class (float/single)
    '*      fClayClass  clay class (float/single)
    '*********************************************************************
    '* Arguments:
    '*      fSand       sand fraction of soil to be classified (float/single)
    '*      fSilt       silt fraction of soil to be classified (float/single)
    '*      fClay       clay fraction of soil to be classified (float/single)
    '*      iNumInt     number of classification intervals to use (integer)
    '*********************************************************************
    '* Returns:  classification string in the format of saXXXsiYYYclZZZ, where
    '*    XXX = sand fraction (000 to 100)
    '*    YYY = silt fraction (000 to 100)
    '*    ZZZ = clay fraction (000 to 100)
    '*********************************************************************


    Dim iFrac As Integer
    Dim fSiltClass As Single
    Dim fClayClass As Single
    Dim fSandClass As Single

    'first check for an error condition
    If fSand + fSilt + fClay <> 100 Then
        soilclass = "Error- soil fractions <> 100"
        GoTo error_SoilClass1
    End If
    DoEvents

    'classify the clay fraction
    iFrac = 100 / iNumInt / 2
    Do While iFrac <= 99.9
        If fClay >= (iFrac - (100 / iNumInt / 2)) And fClay < (iFrac + (100 / iNumInt / 2)) Then
            fClayClass = Left(iFrac, IIf(Len(Str(Int(iFrac))) = 2, 4, 3))
            Exit Do
        End If
        iFrac = iFrac + (100 / iNumInt)
        DoEvents
    Loop

    'classify the silt fraction
    iFrac = 100 / iNumInt / 2
    Do While iFrac <= 99.9
        If fSilt >= (iFrac - (100 / iNumInt / 2)) And fSilt < (iFrac + (100 / iNumInt / 2)) Then
            fSiltClass = Left(iFrac, IIf(Len(Str(Int(iFrac))) = 2, 4, 3))
            Exit Do
        End If
        iFrac = iFrac + (100 / iNumInt)
        DoEvents
    Loop

    'classify the sand fraction
    If fSiltClass + fClayClass > 100 Then
        fSiltClass = (100 - fClayClass)
    End If
    fSandClass = 100 - fClayClass - fSiltClass
    'set the return value
    soilclass = "sa" & IIf(fSandClass <10, "00", IIf(fSandClass >9 And fSandClass <100, "0", "")) &
    fSandClass & "si" & IIf(fSiltClass <10, "00", IIf(fSiltClass >9 And fSiltClass <100, "0", "")) &
```

```
    fSiltClass & "cl" & IIf(fClayClass <10, "00", IIf(fClayClass >9 And fClayClass <100, "0", "")) &
fClayClass
error_SoilClass1:
End Function
```

Appendix 5: Soil Drainage Classification Function

```
Function soter_drainage(drainage1 As String)
    '*********************************************************************
    '* subroutine:  soter_drainage
    '* purpose: returns a hydric status based on the FAO drainage classifications
    '* used in the SOTER database.
    '* author: M. Easter, 22 Dec 2004
    '* Natural Resource Ecology Laboratory, CSU
    '*********************************************************************
    '* Arguments:
    '*      sdrainage   FAO drainage class (string)
    '*********************************************************************
    '* Returns:
    '*   "H" = hydric
    '*   "N" = non-hydric
    '*********************************************************************
    If sdrainage = "E" Then soter_drainage = "N"
    If sdrainage = "W" Then soter_drainage = "N"
    If sdrainage = "M" Then soter_drainage = "N"
    If sdrainage = "I" Then soter_drainage = "N"
    If sdrainage = "P" Then soter_drainage = "N"
    If sdrainage = "S" Then soter_drainage = "H"
    If sdrainage = "V" Then soter_drainage = "H"
End Function
```

Appendix 6: Function defining IPCC soil classifications based on SOTER classifications (I guess this has to be amended to incorporate Niels' suggestions?)

```
Function IPCCSOTER(claf As String) As String
    If Left(claf, 2) = "AC" Then IPCCSOTER = "Low clay activity mineral"
    If Left(claf, 2) = "AL" Then IPCCSOTER = "High clay activity mineral"
    If Left(claf, 2) = "AN" Then IPCCSOTER = "Volcanic"
    If Left(claf, 2) = "AR" Then IPCCSOTER = "Sandy"
    If Left(claf, 2) = "AT" Then IPCCSOTER = "High clay activity mineral"
    If Left(claf, 2) = "CH" Then IPCCSOTER = "High clay activity mineral"
    If Left(claf, 2) = "CL" Then IPCCSOTER = "High clay activity mineral"
    If Left(claf, 2) = "CM" Then IPCCSOTER = "High clay activity mineral"
    If Left(claf, 2) = "FL" Then IPCCSOTER = "High clay activity mineral"
    If Left(claf, 2) = "FR" Then IPCCSOTER = "Low clay activity mineral"
    If Left(claf, 2) = "GL" Then IPCCSOTER = "Aquic"
    If Left(claf, 2) = "GR" Then IPCCSOTER = "High clay activity mineral"
    If Left(claf, 2) = "GY" Then IPCCSOTER = "High clay activity mineral"
    If Left(claf, 2) = "HS" Then IPCCSOTER = "Organic"
    If Left(claf, 2) = "KS" Then IPCCSOTER = "High clay activity mineral"
    If Left(claf, 2) = "LP" Then IPCCSOTER = "High clay activity mineral"
    If Left(claf, 2) = "LV" Then IPCCSOTER = "High clay activity mineral"
    If Left(claf, 2) = "LX" Then IPCCSOTER = "Low clay activity mineral"
    If Left(claf, 2) = "NT" Then IPCCSOTER = "Low clay activity mineral"
    If Left(claf, 2) = "PD" Then IPCCSOTER = "High clay activity mineral"
    If Left(claf, 2) = "PH" Then IPCCSOTER = "High clay activity mineral"
    If Left(claf, 2) = "PL" Then IPCCSOTER = "Aquic"
    If Left(claf, 2) = "PT" Then IPCCSOTER = "Low clay activity mineral"
    If Left(claf, 2) = "PZ" Then IPCCSOTER = "Sandy"
    If Left(claf, 2) = "RG" Then IPCCSOTER = "High clay activity mineral"
    If Left(claf, 2) = "SC" Then IPCCSOTER = "High clay activity mineral"
    If Left(claf, 2) = "SN" Then IPCCSOTER = "High clay activity mineral"
```

```
    If Left(claf, 2) = "VR" Then IPCCSOTER = "High clay activity mineral"
End Function
```

Appendix 7: Additional Information Sources

GEFSOC project web site:
http://www.reading.ac.uk/GEFSOC/index.htm

Natural Resource Ecology Laboratory (NREL), USA:
http://www.nrel.colostate.edu/

GEFSOC Project web site at NREL:
http://www.nrel.colostate.edu/projects/agroecosystems/gefsoc/index.html

Centro de Energia Nuclear na Agricultura (CENA), Brazil :
http://www.cena.usp.br/

Higher Council for Research and Technology/Badia Research and Development Centre (BDRC), Jordan: http://www.badia.gov.jo/

Kenya Soil Survey, Kenya:
http://www.kari.org/

International Institute for Applied Systems Analysis, Austria:
http://www.iiasa.ac.at/

ISRIC- World Soils Information, Netherlands:
http://www.isric.org/

Institut de Recherche pour le Developpement (IRD), France
http://www.ird.fr/

Rothamsted Research Centre, UK
http://www.rothamsted.ac.uk/

Centre for Ecology and Hydrology, UK
http://www.ceh.ac.uk/aboutceh/edinburgh.htm

The University of Aberdeen, UK
http://www.aberdeen.ac.uk/

United Nations Environment Programme
http://www.unep.org/

The Hadley Centre, UK
http://www.metoffice.com/research/hadleycentre/

National Bureau of Soil Survey and Land Use Planning, India
Indian Council of Agricultural Research, Nagpur 440 010, India

Bibliography

[i] Milne, E. On behalf of the GEFSOC Project Team. Towards a generic system for estimating soil carbon stocks and changes at the regional and national scale. Presented at 'Impacts of changes of land use and manageent on carbon stocks and turnover in the tropics' The Institute of geography, University of Copenhagen, Denmark. August 2004.

[ii] Parton, W.J., D.S. Ojima, D.S. Schimel and T.G.F. Kittel. 1992.Development of simplified ecosystem models for applications in earth system studies: The CENTURY experience. Pages 281-302 in D.S. Ojima (ed.) Modeling the Earth System. Proceedings from the 1990 Global Change Institute on Earth System Modeling, Snowmass, Colorado, 16-27 July 1990.

[iii] Paustian, K., W.J. Parton and J. Persson. 1992. Modeling soil organic matter in organic-amended and nitrogen-fertilized long-term plots. Soil Science Society of America Journal 56:476-488

[iv] Coleman K. and Jenkinson D.S. (1995) RothC-26 3. A model for the turnover of carbon in soil: model description and users guide. ISBN 09514456 69.

[v] Penman, Jim, Michael Gytarsky, Taka Hiraishi, Thelma Krug, Dina Kruger, Riitta Pipatti, Leandro Buendia, Kyoko Miwa, Todd Ngara, Kiyoto Tanabe and Fabian Wagner (Eds). 2003. Good Practice Guidance for Land Use, Land-Use Change and Forestry. The Intergovernmental Panel on Climate Change (IPCC), C/o Institute for Global Environmental Strategies, 2108 -11, Kamiyamaguchi, Hayama, Kanagawa, Japan, 240-0115, Fax: (81 46) 855 3808, http://www.ipcc-nggip.iges.or.jp

[vi] Van Engelen VWP and Wen TT (1995). Global and National Soils and Terrain Databases (SOTER): Procedures Manual (rev. ed.). (Published also as FAO World Soil Resources Report No. 74), UNEP, IUSS, ISRIC and FAO, Wageningen

[vii] Paustian, K., M.J. Easter, S.W. Williams and K. Killian. 2005. Unpublished data.

[viii] Gutmann, Myron, Parton, William, Ojima, Denis, Williams, Stephen, Easter, Mark. (2005, accepted). Human population and environment in the U.S. Great Plains. Manuscript submitted to *Ecological Modeling*.

[ix] Ibid.

[x] Metherell, Alister K., Laura A. Harding, C. Vernon Cole and William J. Parton. CENTURY Soil Organic Matter Model Environment. Technical Documentation for Agroecosystem Version 4.0. Great Plains System Research Unit, Technical Report No. 4, USDA-ARS, Fort Collins, Colorado. USA.

[xi] Ibid.

[xii] PRISM dataset. 2005. Spatial Climate Analysis Service, Oregon Climate Service, http://www.ocs.orst.edu/prism/